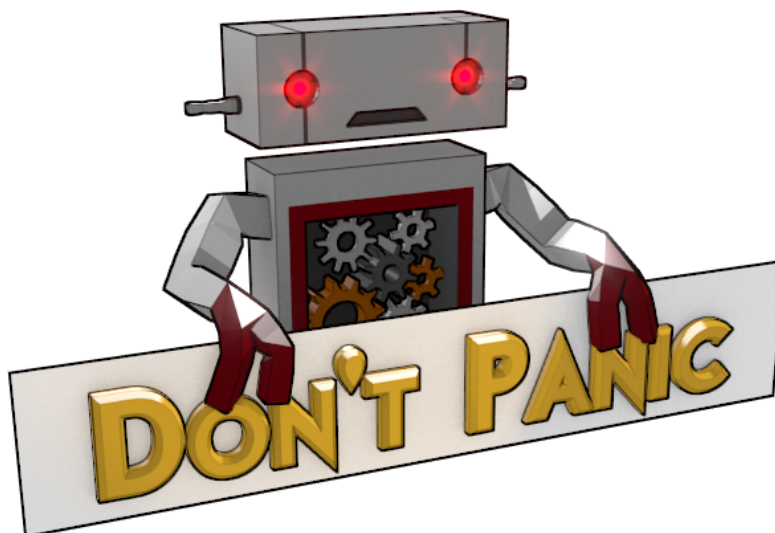


# DON'T PANIC

3DMob: Grafica 3D su device mobili

---



## Norme di progetto

### Informazioni sul documento

---

Versione	1.2.0
Redazione	Basaglia Mattia Rampazzo Federico Pezzutti Marco Sciarrone Riccardo
Verifica	Busato Luca Lain Daniele
Responsabile	Cesarato Fabio
Uso	Interno
Lista di distribuzione	Don't Panic

### Descrizione

Documento riguardante le norme stabilite dal gruppo Don't Panic per la realizzazione di 3DMob



## Diario delle modifiche

Descrizione modifica	Autore	Data	Versione
Approvazione documento	Cesarato Fabio	2012-11-25	1.2.0
Verifica documento	Lain Daniele	2012-11-24	1.1.1
Verifica documento	Busato Luca	2012-11-23	1.1.0
Modificata sezione versionamento	Pezzutti Marco	2012-11-23	1.0.11
Stesura ambiente di lavoro, progettazione, introduzione, codifica, glossario	Rampazzo Federico	2012-11-22	1.0.10
Stesura ambiente di lavoro	Rampazzo Federico	2012-11-22	1.0.9
Analisi dei requisiti	Sciarrone Riccardo	2012-11-22	1.0.8
Aggiunto albero del repository	Sciarrone Riccardo	2012-11-22	1.0.7
Correzioni norme di comunicazione e riunione	Pezzutti Marco	2012-11-22	1.0.6
Stesura norme di comunicazione e riunione	Sciarrone Riccardo	2012-11-22	1.0.5
Stesura norme di ciclo di vita dei documenti	Pezzutti Marco	2012-11-21	1.0.4
Descrizione software per il Project Management e norme utilizzo sistema di ticketing	Basaglia Mattia	2012-11-21	1.0.3
Stesura norme repository	Sciarrone Riccardo	2012-11-21	1.0.2
Stesura norme tipografiche	Basaglia Mattia	2012-11-21	1.0.1
Creazione scheletro del documento e stesura parziale di documenti	Rampazzo Federico	2012-11-20	1.0.0



## Indice

<b>1</b>	<b>Introduzione</b>	<b>1</b>
1.1	Scopo del documento . . . . .	1
1.2	Scopo del Prodotto . . . . .	1
1.3	Glossario . . . . .	1
1.4	Riferimenti . . . . .	1
1.4.1	Informativi . . . . .	1
<b>2</b>	<b>Comunicazioni</b>	<b>2</b>
2.1	Comunicazioni esterne . . . . .	2
2.2	Comunicazioni interne . . . . .	2
2.3	Composizione e-mail . . . . .	2
2.3.1	Destinatario . . . . .	2
2.3.2	Mittente . . . . .	3
2.3.3	Oggetto . . . . .	3
2.3.4	Corpo . . . . .	3
2.3.5	Allegati . . . . .	3
<b>3</b>	<b>Riunioni</b>	<b>4</b>
3.1	Frequenza . . . . .	4
3.2	Convocazione riunione . . . . .	4
3.2.1	Interna . . . . .	4
3.3	Svolgimento riunione . . . . .	4
3.3.1	Esterna . . . . .	4
3.4	Verbale . . . . .	5
3.4.1	Riunione interna . . . . .	5
3.4.2	Riunione esterna . . . . .	5
<b>4</b>	<b>Documenti</b>	<b>6</b>
4.1	Template . . . . .	6
4.2	Struttura del documento . . . . .	6
4.2.1	Prima pagina . . . . .	6
4.2.2	Diario delle modifiche . . . . .	6
4.2.3	Indici . . . . .	7
4.2.4	Formattazione generale delle pagine . . . . .	7
4.3	Norme tipografiche . . . . .	7
4.3.1	Punteggiatura . . . . .	7
4.3.2	Stile di testo . . . . .	7
4.3.3	Composizione del testo . . . . .	8
4.3.4	Formati . . . . .	8
4.3.5	Sigle . . . . .	9
4.4	Componenti grafiche . . . . .	10
4.4.1	Tabelle . . . . .	10
4.4.2	Immagini . . . . .	10
4.5	Classificazione documenti . . . . .	10
4.5.1	Documenti formali . . . . .	10
4.5.2	Documenti informali . . . . .	10
4.6	Versionamento . . . . .	10



4.7	Ciclo di vita . . . . .	11
<b>5</b>	<b>Analisi dei requisiti</b>	<b>12</b>
5.1	Studio di Fattibilità e Analisi dei Rischi . . . . .	12
5.2	Analisi dei requisiti . . . . .	12
5.2.1	Classificazione dei requisiti . . . . .	12
5.2.2	Modellazione concettuale del sistema e Allocazione . . . . .	13
5.2.3	Tracciamento . . . . .	13
<b>6</b>	<b>Codifica dei file e documentazione</b>	<b>14</b>
6.1	Codifica e convenzioni . . . . .	14
6.2	Documentazione . . . . .	14
<b>7</b>	<b>Glossario</b>	<b>15</b>
<b>8</b>	<b>Protocollo per lo sviluppo dell'applicazione</b>	<b>16</b>
8.1	Creare un nuovo progetto . . . . .	16
8.2	Creazione ticket . . . . .	16
8.2.1	Ticket di pianificazione . . . . .	17
8.2.2	Ticket di realizzazione e controllo . . . . .	17
8.2.3	Ticket di verifica . . . . .	18
8.2.4	Dipendenze temporali . . . . .	18
8.3	Aggiornamento ticket . . . . .	19
8.3.1	Ticket di pianificazione . . . . .	19
8.3.2	Ticket di realizzazione e controllo . . . . .	19
8.3.3	Ticket di verifica . . . . .	20
8.4	Consigli di utilizzo . . . . .	20
8.4.1	Pagina personale . . . . .	20
8.4.2	Visualizzare segnalazioni . . . . .	20
<b>9</b>	<b>Ambiente di lavoro</b>	<b>21</b>
9.1	Coordinamento . . . . .	21
9.1.1	Server dedicato . . . . .	21
9.1.2	Dropbox . . . . .	23
9.1.3	Google Drive e Calendar . . . . .	23
9.2	Ambiente documentale . . . . .	23
9.2.1	Pianificazione . . . . .	23
9.2.2	Stesura documenti . . . . .	24
9.3	Ambiente di sviluppo . . . . .	26
9.3.1	Framework . . . . .	27
9.4	Ambiente di verifica . . . . .	27
9.4.1	Documenti . . . . .	27
9.4.2	Codice . . . . .	28



## Elenco delle figure

1	Strumento tracciamento requisiti - screenshot . . . . .	26
---	---	----



## 1 Introduzione

### 1.1 Scopo del documento

Questo documento definisce le norme che i membri del gruppo Don't Panic adotteranno nello svolgimento del progetto "3DMob".

Tutti i membri sono tenuti a leggere il documento e a seguire le norme per migliorare l'uniformità del materiale prodotto, migliorare l'efficienza e ridurre il numero di errori. In particolare verranno definite norme riguardanti:

- Interazioni tra membri del gruppo;
- Stesura 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'applicazione in grado di convertire file prodotti da programmi di grafica 3D in file in formato JSON<sub>G</sub> in grado di essere visualizzati su dispositivi mobile senza perdita di informazione. L'obiettivo è quello di semplificare il workflow attuale necessario a rendere compatibili i file.

### 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 v1.2.0*.

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

### 1.4 Riferimenti

#### 1.4.1 Informativi

- Specifiche UTF-8<sub>G</sub>:  
<http://www.unicode.org/versions/Unicode6.1.0/ch03.pdf>;
- Licenza GNU LGPL<sub>G</sub> 2.1: <http://www.gnu.org/licenses/lgpl-2.1.html>;
- Qt Coding Conventions<sub>G</sub>:  
<http://qt-project.org/wiki/Coding-Conventions>;
- Qt<sub>G</sub> Coding Style: [http://qt-project.org/wiki/Qt\\_Coding\\_Style](http://qt-project.org/wiki/Qt_Coding_Style);
- ISO<sub>G</sub> 8601: [http://it.wikipedia.org/wiki/ISO\\_8601](http://it.wikipedia.org/wiki/ISO_8601);
- Jenkins: [http://en.wikipedia.org/wiki/Jenkins\\_\(software\)](http://en.wikipedia.org/wiki/Jenkins_(software));
- Piano di Progetto: *Piano di Progetto v1.2.0*;
- Piano di Qualifica: *Piano di Qualifica v1.2.0*.



## 2 Comunicazioni

### 2.1 Comunicazioni esterne

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

[dont.panic.swe@gmail.com](mailto:dont.panic.swe@gmail.com)

Tale indirizzo deve essere l'unico canale di comunicazione esistente tra il gruppo di lavoro e l'esterno. Solo il *Responsabile di Progetto* ha accesso 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 inoltrare, qualora sia necessario, il messaggio alla mailing list<sub>G</sub> ufficiale.

### 2.2 Comunicazioni interne

Per le comunicazioni interne è stata creata una mailing list<sub>G</sub>:

[ntp@myopenproject.it](mailto:ntp@myopenproject.it)

I membri del gruppo sono quindi tenuti ad utilizzare l'indirizzo [ntp@myopenproject.it](mailto:ntp@myopenproject.it) per comunicare tra loro. Ogni componente è quindi costantemente informato sullo scambio di informazioni interne. Questo strumento va utilizzato correttamente e solo per questioni inerenti al progetto. Un redirect provvederà ad inoltrare le mail destinate a [ntp@myopenproject.it](mailto:ntp@myopenproject.it) al fine di mantenere uno storico, consultabile online mediante la webmail<sub>G</sub>:

<http://webmail.myopenproject.it>

Con lo scopo di facilitare la comunicazione tra i membri del gruppo, viene adottato l'uso di sistemi di instant messaging<sub>G</sub> e videoconferenza quali Skype e Google Plus, con l'obbligo di redigere un verbale da inviare alla mailing list una volta terminata la conversazione. La verbalizzazione ha l'obiettivo di tenere traccia di ogni argomento discusso all'interno del gruppo, in quanto una comunicazione verbale non documentata non è accettabile per il corretto svolgimento del progetto.

È sconsigliato l'uso del telefono e degli SMS, ma in caso di necessità è possibile ricorrere a tali strumenti. Si chiede che la conversazione venga documentata tramite mail nel caso vengano prese decisioni o emergano contenuti utili allo sviluppo del progetto.

### 2.3 Composizione e-mail

In questo paragrafo viene descritta la struttura che deve avere un messaggio sia per una comunicazione interna che esterna.

#### 2.3.1 Destinatario

- **Interno:** l'unico indirizzo utilizzabile è [ntp@myopenproject.it](mailto:ntp@myopenproject.it);
- **Esterno:** può variare a seconda che ci si debba riferire al Proponente, al Prof. Vardanega Tullio o al Prof. Cardin Riccardo.



### 2.3.2 Mittente

- **Interno:** l'indirizzo personale di chi scrive il messaggio;
- **Esterno:** l'unico indirizzo utilizzabile è [dont.panic.swe@gmail.com](mailto:dont.panic.swe@gmail.com) e deve essere usato solamente dal *Responsabile di Progetto*.

### 2.3.3 Oggetto

L'oggetto deve essere chiaro ed esaustivo, possibilmente stringato e non confondibile con altri preesistenti. Nel caso si debba comporre un messaggio di risposta vi è l'obbligo di aggiungere la particella "Re:" all'inizio dell'oggetto per essere in grado di distinguere il livello di risposta; se si dovesse trattare di un inoltro si deve usare invece la particella "I:". In ogni caso, l'oggetto di una comunicazione, una volta avviata, non deve mai essere cambiato.

### 2.3.4 Corpo

Il corpo di un messaggio deve contenere tutte le informazioni necessarie a rendere facilmente comprensibile l'argomento trattato a tutti i membri del gruppo.

Se alcune parti del messaggio hanno uno o più destinatari specifici, il loro nome dovrà essere aggiunto all'inizio del paragrafo tramite la seguente segnatura: *@destinatario*.

In caso di risposta od inoltro del messaggio, il contenuto aggiunto deve essere sempre messo in testa (per non costringere gli altri membri a dover scorrere tutta la mail). Si consiglia di non cancellare il resto del messaggio, per consentire una visione completa della discussione.

### 2.3.5 Allegati

Viene permesso l'uso di allegati qualora ve ne sia la necessità. Essi possono essere usati ad esempio per allegare il verbale di una comunicazione via instant messaging<sub>G</sub> tra componenti del gruppo di lavoro o per il resoconto di un incontro con il proponente o con il committente.





## 3 Riunioni

### 3.1 Frequenza

Le riunioni del gruppo di lavoro avranno una frequenza almeno quindicinale.

### 3.2 Convocazione riunione

#### 3.2.1 Interna

Il *Responsabile di Progetto* ha il compito di convocare le riunioni generali, ovvero le riunioni in cui vengono convocati tutti i membri del gruppo, ed eventualmente di valutare se anticipare la naturale scadenza della successiva riunione.

Qualora ve ne sia la necessità, qualsiasi componente può richiedere la convocazione di una riunione, ma tale richiesta deve essere inoltrata al *Responsabile di Progetto* che può decidere se accettarla o respingerla. È inoltre possibile e auspicabile che possano essere necessarie riunioni tra specifici membri: ad esempio, nella fase di progettazione può essere utile la collaborazione tra *Progettista* e *Analista*, senza richiedere la presenza di persone non direttamente coinvolte, che verranno comunque informate delle decisioni prese tramite verbale inviato mediante posta elettronica.

Il responsabile deve convocare l'assemblea con almeno tre giorni di preavviso attraverso l'invio una email a [ntp@myopenproject.it](mailto:ntp@myopenproject.it) contenente

- **Oggetto:** Convocazione riunione n. X, dove X indica il numero crescente di riunioni effettuate;
- **Corpo:**
  - **Data:** data e ora prevista;
  - **Luogo:** luogo previsto;
  - **Tipo:** ordinaria/straordinaria;
  - **Ordine del giorno:** elenco numerato delle varie voci da esaminare.

Ogni componente del gruppo deve rispondere al messaggio nel minor tempo possibile confermando la presenza o motivando un'eventuale assenza. In caso di mancata risposta il *Responsabile di Progetto* ha il dovere di contattare anche telefonicamente colui che non ha fornito una risposta in tempo utile. Il *Responsabile di Progetto*, una volta ricevute le risposte e verificata la presenza o assenza dei membri richiesti, può decidere se confermare, annullare o spostare la riunione, per permettere la partecipazione agli eventuali assenti. La conferma, l'annullamento e lo spostamento dell'assemblea devono essere effettuati tramite email.

### 3.3 Svolgimento riunione

#### 3.3.1 Esterna

All'apertura della riunione, verificata la presenza dei membri previsti, viene scelto un segretario che avrà il compito di annotare ogni argomento trattato e di redigere il verbale dell'assemblea, che dovrà poi essere inviato ai restanti componenti del gruppo.

Tutti i partecipanti devono osservare un comportamento consono al miglior svolgimento della riunione e al raggiungimento degli obiettivi della stessa. Il segretario deve inoltre controllare che venga seguito l'ordine del giorno in modo da non tralasciare alcun punto.



## 3.4 Verbale

### 3.4.1 Riunione interna

Il verbale di riunione interna è un documento interno informale che consente di tracciare gli argomenti discussi durante la riunione.

Verrà redatto dal segretario della riunione, ruolo scelto di volta in volta a rotazione tra i presenti.

Si consiglia di redigere il verbale di una riunione interna in un documento di testo da condividere con Google Drive<sub>G</sub>, in modo da rendere sempre disponibile il contenuto dello stesso.

Il verbale deve anche essere inviato a tutto il gruppo di lavoro tramite posta elettronica allegando il documento ad un messaggio di risposta all'email di convocazione dell'assemblea.

### 3.4.2 Riunione esterna

In caso di riunione con il committente od il proponente, 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. Vi è quindi l'obbligo di seguire il sopraccitato schema per creare il verbale e di inviare il documento come allegato alla mailing list<sub>G</sub> in risposta all'email di convocazione dell'assemblea.



## 4 Documenti

Questo capitolo descrive tutte le convenzioni scelte ed adottate da Don't Panic riguardo alla stesura, verifica e approvazione della documentazione da produrre.

### 4.1 Template

Per agevolare la redazione della documentazione è stato creato un template  $\text{\LaTeX}$  contenente tutte le impostazioni stilistiche e grafiche citate in questo documento. Tale modello si può trovare nel repository<sub>G</sub> in `documents/template`.

### 4.2 Struttura del documento

#### 4.2.1 Prima pagina

Ogni documento è caratterizzato da una prima pagina che contiene le seguenti informazioni sul documento:

- Nome del gruppo;
- Nome del progetto;
- Logo del gruppo;
- Titolo del documento;
- Versione del documento;
- Cognome e nome dei redattori del documento;
- Cognome e nome dei verificatori del documento;
- Cognome e nome del responsabile approvatore del documento;
- Destinazione d'uso del documento;
- Lista di distribuzione del documento;
- Una breve descrizione del documento.

#### 4.2.2 Diario delle modifiche

La seconda pagina di ogni documento contiene il diario delle modifiche del documento. Ogni riga del diario delle modifiche contiene:

- Un breve sommario delle modifiche svolte;
- Cognome e nome dell'autore;
- Data della modifica;
- Versione del documento dopo la modifica.

La tabella è ordinata per data in ordine decrescente, in modo che la prima riga corrisponda alla versione attuale del documento.



### 4.2.3 Indici

In ogni documento è presente un indice delle sezioni, un indice delle figure e un indice delle tabelle. Nel caso non siano presenti figure o tabelle i rispettivi indici verranno omessi.

### 4.2.4 Formattazione generale delle pagine

L'intestazione di ogni pagina contiene:

- Logo del gruppo;
- Nome del gruppo;
- Nome del progetto;
- Sezione corrente del documento.

A piè di pagina invece è presente:

- Nome e versione del documento;
- Pagina corrente nel formato  $N$  di  $T$  dove  $N$  è il numero di pagina corrente e  $T$  è il numero di pagine totali.

## 4.3 Norme tipografiche

Questa sezione racchiude le convenzioni riguardanti tipografia, ortografia e uno stile uniforme per tutti i documenti.

### 4.3.1 Punteggiatura

- **Parentesi:** il testo racchiuso tra parentesi non deve aprirsi o chiudersi con un carattere di spaziatura e non deve terminare con un carattere di punteggiatura;
- **Punteggiatura:** un carattere di punteggiatura non deve mai seguire un carattere di spaziatura;
- **Lettere maiuscole:** le lettere maiuscole vanno poste solo dopo il punto, il punto di domanda, il punto esclamativo e all'inizio di ogni elemento di un elenco puntato, oltre che dove previsto dalla lingua italiana. È inoltre utilizzata l'iniziale maiuscola nel nome del team, del progetto, dei documenti, dei ruoli di progetto, delle fasi di lavoro e nelle parole Proponente e Committente.

### 4.3.2 Stile di testo

- **Corsivo:** il corsivo deve essere utilizzato nei seguenti casi:
  - **Citazioni:** quando si deve citare una frase questa va scritta in corsivo;
  - **Abbreviazioni:** quando possibile si deve preferire una parola completa ad un'abbreviazione;
  - **Nomi particolari:** il corsivo deve essere utilizzato quando si parla di figure particolari (es. *Progettista*);



- **Documenti:** il corsivo deve essere utilizzato quando si parla di documenti (es. *Glossario*);
- **Altri casi:** in altre situazione, il corsivo va utilizzato per mettere in rilievo passaggi o parole significativi, evidenziare riferimenti ai documenti interni o esterni.
- **Grassetto:** il grassetto può essere utilizzato nei seguenti casi:
  - **Elenchi puntati:** in questi casi può essere utilizzato il grassetto per evidenziare il concetto sviluppato nella continuazione del punto;
  - **Altri casi:** per evidenziare particolari passaggi o parole chiave.
- **Monospace<sub>G</sub>:** il carattere monospace<sub>G</sub> serve per formattare il testo contenente parti di codice, nomi di classi, comandi, indirizzi web e percorsi;
- **Maiuscolo:** l'utilizzo di parole completamente in maiuscolo è riservato solo agli acronimi o alle macro  $\text{\LaTeX}$  riportate nei documenti;
- **$\text{\LaTeX}$ :** ogni riferimento a  $\text{\LaTeX}$  va scritto utilizzando il comando  $\text{\LaTeX}$ .

#### 4.3.3 Composizione del testo

- **Elenchi puntati:** ogni punto dell'elenco deve terminare con un punto e virgola, tranne l'ultimo che deve terminare con un punto. La prima parola deve avere la lettera maiuscola, a meno di casi particolari (es. nome di un file);
- **Note a piè di pagina:** ogni nota dovrà cominciare con l'iniziale della prima parola maiuscola e non deve essere preceduta da alcun carattere di spaziatura. Ogni nota deve terminare con un punto.

#### 4.3.4 Formati

- **Percorsi:** per gli indirizzi email e web completi deve essere utilizzato il comando  $\text{\LaTeX}$   $\text{\url}$ , mentre per gli indirizzi relativi va usato il formato monospace<sub>G</sub>;
- **Date:** le date presenti nei documenti devono seguire la notazione definita dallo standard ISO<sub>G</sub> 8601:2004:

$$AAAA - MM - GG$$

dove:

- *AAAA*: rappresenta l'anno scritto utilizzando quattro cifre;
- *MM*: rappresenta il mese scritto utilizzando due cifre;
- *GG*: rappresenta il giorno scritto utilizzando due cifre.
- **Ruoli di progetto:** è necessario riferirsi ai ruoli di progetto mediante il comando  $\text{\LaTeX}$   $\text{\role{Nome del ruolo}}$ , che garantisce la corretta scrittura, con la prima lettera di ogni parola che non sia una preposizione maiuscola (es. *Amministratore*);



- **Riferimenti ai documenti:** è necessario riferirsi ai documenti mediante il comando  $\text{\LaTeX \doc{Nome del documento}}$ , che garantisce la corretta scrittura, con la prima lettera di ogni parola che non sia una preposizione maiuscola (es. *Revisione dei Requisiti*).

Nel caso sia necessario riferirsi alla versione più recente del documento, è necessario l'uso dei comandi  $\text{\LaTeX}$  appositamente predisposti, quali ad esempio  $\text{\LaTeX \NormeDiProgetto}$ , che garantisce il corretto riferimento (es. *Norme di Progetto v1.2.0*);

- **Nomi dei file:** nel caso in cui ci si riferisca a un file senza utilizzare il suo percorso completo si deve utilizzare il formato  $\text{\LaTeX monospace}_G$ ;
- **Revisioni:** è necessario riferirsi alle revisioni programmate con il comando  $\text{\LaTeX \rev{Nome Revisione}}$ , che garantisce la corretta scrittura, con la prima lettera di ogni parola che non sia una preposizione maiuscola (es. **Revisione dei Requisiti**);
- **Fasi del progetto:** è necessario riferirsi alle fasi del progetto con il comando  $\text{\LaTeX \fasi{Nome Fase}}$ , che garantisce la corretta scrittura, con la prima lettera di ogni parola che non sia una preposizione maiuscola (es. **Verifica e Validazione**);
- **Nomi propri:** l'utilizzo dei nomi propri dei membri del team deve seguire la forma "Cognome Nome";
- **Nome del gruppo:** ci si riferirà al gruppo solo come "Don't Panic". Per la corretta scrittura è definita la macro  $\text{\LaTeX \GRUPPO}$ ;
- **Nome del proponente:** ci si riferirà al proponente come "Mentis srl" o con "Proponente". Per la corretta scrittura è definita la macro  $\text{\LaTeX \PROPONENTE}$ ;
- **Nome del committente:** ci si riferirà al committente come "Prof. Vardanega Tullio" o con "Committente". Per la corretta scrittura è definita la macro  $\text{\LaTeX \COMMITTENTE}$ ;
- **Nome del progetto:** ci si riferirà al progetto solo come "3DMob"; per la corretta scrittura è definita la macro  $\text{\LaTeX \PROGETTO}$ .

#### 4.3.5 Sigle

Le sigle dei documenti e delle revisioni potranno essere utilizzate esclusivamente all'interno di tabelle o diagrammi. In particolare sono previste le seguenti sigle:

- **AdR** = Analisi dei Requisiti;
- **GL** = Glossario;
- **NdP** = Norme di Progetto;
- **PdP** = Piano di Progetto;
- **PdQ** = Piano di Qualifica;
- **SdF** = Studio di Fattibilità;



- **ST** = Specifica Tecnica;
- **RA** = Revisione di Accettazione;
- **RP** = Revisione di Progettazione;
- **RQ** = Revisione di Qualifica;
- **RR** = Revisione dei Requisiti.

## 4.4 Componenti grafiche

### 4.4.1 Tabelle

Ogni tabella presente all'interno dei documenti dev'essere accompagnata da una didascalia, in cui deve comparire un numero identificativo incrementale per la tracciabilità della stessa all'interno del documento.

Per aumentare la leggibilità, in caso di tabelle contenenti molti numeri, le celle con molteplici valori uguali possono essere lasciate vuote. Tale scelta deve essere riportata all'inizio del documento che ne fa uso, specificando il valore che non verrà riportato.

### 4.4.2 Immagini

Le immagini presenti all'interno dei documenti devono essere nel formato Scalable Vector Graphics (SVG<sub>G</sub>). In questo modo si garantisce una maggior qualità dell'immagine in caso di ridimensionamento. Per consentire l'inclusione delle immagini nei documenti, le immagini dovranno essere convertite nel formato PDF<sub>G</sub>. Qualora non sia possibile salvare le immagine in formato vettoriale è preferito il formato Portable Network Graphics (PNG<sub>G</sub>).

## 4.5 Classificazione documenti

### 4.5.1 Documenti formali

Un documento viene definito formale quando viene approvato dal *Responsabile di Progetto* ed è quindi pronto ad essere diffuso ai richiedenti. Questo momento coincide con una revisione con il committente. Per giungere a questo stato ogni documento deve aver seguito il percorso di verifica e validazione descritto nel *Piano di Qualifica v1.2.0* e nel paragrafo 4.7 riguardante il ciclo di vita dei documenti.

### 4.5.2 Documenti informali

Un documento è ritenuto informale dal momento della sua creazione fino a quando non viene approvato dal *Responsabile di Progetto* ed è quindi da considerarsi esclusivamente ad uso interno.

## 4.6 Versionamento

La documentazione prodotta deve essere corredata del numero di versione attuale tramite la seguente codifica:

$$vX.Y.Z$$

dove:



- **X:** indica il numero crescente di uscite formali del documento
  1. Fase di **Analisi** che si conclude con la **Revisione dei Requisiti**;
  2. Fase di **Analisi di Dettaglio** che si conclude con l'ingresso nella fase successiva;
  3. Fase di **Progettazione Architetture** che si conclude con la **Revisione di Progettazione**;
  4. Fase di **Progettazione di Dettaglio e Codifica** che si conclude con la **Revisione di Qualifica**;
  5. Fase di **Verifica e Validazione** che si conclude con la **Revisione di Accettazione**.
- **Y:** indica il numero crescente di modifiche sostanziali al documento
  0. Fase di stesura del documento;
  1. Fase di verifica del documento;
  2. Approvazione del documento.

- **Z:** indica il numero crescente di modifiche minori effettuate nelle varie fasi.

Ogniquale sia necessaria la citazione di una versione specifica di un documento, essa deve comprendere sia il nome che il numero di versione aderente al formato:

*Nome Documento vX.Y.Z*

Il formato da applicare ai nomi dei file alla loro creazione invece è:

NomeDocumento\_vX.Y.Z.pdf

## 4.7 Ciclo di vita

Ogni documento prodotto segue un preciso iter che scandisce le fasi in cui si trova in ogni istante. Un documento può trovarsi in tre stati diversi:

- **In lavorazione:** un documento entra in questa fase nel momento della sua creazione, e qui vi rimane per tutto il periodo necessario alla sua realizzazione, o per eventuali successive modifiche;
- **Da verificare:** una volta che il documento viene ultimato, esso deve essere preso in consegna dai verificatori che avranno il compito di rilevare e correggere eventuali errori e/o imprecisioni sintattici e semantici;
- **Approvato:** ogni documento, una volta ultimata la fase di verifica, deve essere approvato dal *Responsabile di Progetto*. L'approvazione sancisce lo stato finale del documento per la data versione.

Durante la sua vita, ogni documento può attraversare ogni fase più di una volta: nel momento in cui un documento approvato necessita di una revisione formale, esso inizia nuovamente il ciclo che, al suo termine, porterà ad una nuova versione dello stesso.





## 5 Analisi dei requisiti

### 5.1 Studio di Fattibilità e Analisi dei Rischi

Alla pubblicazione dei capitolati, è compito del *Responsabile di Progetto* convocare quante riunioni saranno ritenute necessarie per consentire il confronto tra i membri del gruppo su ogni capitolato. Queste riunioni forniranno agli *Analisti* una base riguardante le conoscenze e preferenze di ogni membro del gruppo. È compito degli *Analisti*, sulla base di quanto deciso, redigere uno *Studio di Fattibilità* dei capitolati basandosi su:

- **Dominio tecnologico e applicativo:** conoscenza delle tecnologie richieste, esperienze precedenti con le problematiche poste dal capitolato, conoscenza del dominio applicativo;
- **Rapporto Costi/Benefici:** competitori e prodotti simili già presenti sul mercato, quantità di requisiti obbligatori, costo della realizzazione rapportato al risultato previsto;
- **Individuazione dei rischi:** Comprensione dei punti critici della realizzazione, individuazione di eventuali lacune tecniche o di conoscenza del dominio applicativo dei membri del gruppo, analisi delle difficoltà nell'individuazione dei requisiti e loro verificabilità.

Un'ultima riunione, a *Studio di Fattibilità* concluso, presenterà la decisione finale sul capitolato scelto.

### 5.2 Analisi dei requisiti

La stesura del documento denominato *Analisi dei Requisiti* è compito degli *Analisti*, e si divide nelle fasi illustrate in questa sezione.

#### 5.2.1 Classificazione dei requisiti

È compito degli *Analisti* stilare una lista dei requisiti emersi dal capitolato e da eventuali riunioni con il proponente. Questi requisiti dovranno essere classificati per tipo e importanza, utilizzando la seguente codifica:

R[importanza][tipo][codice]

- **Importanza** può assumere i seguenti valori:

0. Requisito obbligatorio;
1. Requisito desiderabile;
2. Requisito opzionale.

L'uso di numeri permette di ordinare facilmente i requisiti per importanza;

- **Tipo** può assumere i seguenti valori:

- F: Funzionale;
- Q: Di qualità;
- P: Prestazionale;



V: Vincolo.

- **Codice** è il codice univoco di ogni requisito espresso in modo gerarchico.

Ogni requisito è poi esplicito nel seguente modo:

- **Relazioni** di dipendenza con altri requisiti;
- **Descrizione** sintetica ma chiara del requisito.

Tutti i requisiti vanno inseriti in Tracy, software di tracciamento appositamente creato, descritto nel *Piano di Qualifica v1.2.0*, che provvederà alla generazione del codice identificativo univoco.

### 5.2.2 Modellazione concettuale del sistema e Allocazione

Successivamente al tracciamento dei requisiti emersi dal capitolato e da eventuali riunioni, si procede all'analisi dei casi d'uso, denominati nelle sezioni seguenti anche come *use case* o con l'acronimo UC. È richiesta agli *Analisti* l'identificazione dei casi d'uso, procedendo dal generale al particolare, e l'inserimento degli stessi nel software di tracciamento Tracy.

Di ogni caso d'uso è richiesto l'inserimento nel software di:

- **Titolo** sintetico del caso d'uso;
- **Attori** principali e secondari;
- **Descrizione** chiara e dettagliata del caso d'uso;
- **Precondizione** del caso d'uso;
- **Flusso principale degli eventi** specificando per ogni evento titolo, descrizione, attori coinvolti e se l'evento è descritto nel dettaglio da un altro caso d'uso;
- **Scenari alternativi** specificando per ogni scenario alternativo titolo, descrizione, attori coinvolti e se lo scenario è descritto nel dettaglio da un altro caso d'uso;
- **Postcondizione** del caso d'uso;
- **Requisiti** dedotti dal caso d'uso.

Il caso d'uso dovrà essere accompagnato da un grafico riassuntivo in UML<sub>G</sub> 2.x, titolato come il caso d'uso. È compito del software di tracciamento titolare il caso d'uso con un codice univoco gerarchico, nella forma:

UC[codice univoco del padre].[codice progressivo di livello]

Il codice progressivo può includere diversi livelli di gerarchia separati da un punto. Il software provvederà a fornire le tabelle di tracciamento e il codice L<sup>A</sup>T<sub>E</sub>X dei casi d'uso, formattati secondo convenzione.

### 5.2.3 Tracciamento

Sebbene il software di tracciamento sia provvisto di strumenti automatici atti allo scopo, spetta agli *Analisti* controllare la corrispondenza biunivoca di ogni requisito con una o più fonti. Le fonti di requisiti sono: casi d'uso, verbale e capitolato.

In caso di anomalie è necessario correggerle e segnalarle, così da migliorare il software di tracciamento.



## 6 Codifica dei file e documentazione

### 6.1 Codifica e convenzioni

Tutti i file contenenti codice o documentazione dovranno essere in codifica UTF-8<sub>G</sub> senza BOM<sub>G</sub>. Verrà usato LF<sub>G</sub> (U+000A) per andare a capo.

Gli sviluppatori dovranno attenersi alle Qt Coding Conventions<sub>G</sub> e seguire le convenzioni stilistiche definite dal Qt<sub>G</sub> Coding Style.

È ammessa la possibilità di effettuare modifiche alle convenzioni stabilite in seguito ad una decisione del *Responsabile di Progetto*.

L'unica lingua ammessa per i nomi di variabili, classi e funzioni è l'inglese.

### 6.2 Documentazione

I file contenenti codice dovranno essere provvisti di un'intestazione contenente:

```
/*!  
 * \file Nome del file  
 * \author Autore (indirizzo email dell'autore)  
 * \date Data di creazione  
 * \brief Breve descrizione del file  
 *  
 * Descrizione dettagliata del file  
 */
```

Prima di ogni classe dovrà essere presente un commento contenente:

```
/*!  
 * \class Nome della classe  
 * \brief Breve descrizione della classe  
 */
```

Prima di ogni metodo dovrà essere presente un commento contenente:

```
/*!  
 * \brief Breve descrizione della funzione  
 * \param Nome del primo parametro  
 * \param Nome del secondo parametro  
 * \return Valore ritornato dalla funzione  
 */
```



## 7 Glossario

Il Glossario conterrà tutte le parole presenti negli altri documenti che fanno parte del contesto dell'applicazione o che possono essere fraintese. Le definizioni, presentate in ordine alfabetico, dovranno essere concise e comprensibili.

I termini verranno inseriti nel glossario parallelamente al processo di stesura degli altri documenti, in modo da limitare l'errore umano. È preferibile inserire un termine inizialmente privo di definizione, piuttosto che rimandare la stesura del glossario.



## 8 Protocollo per lo sviluppo dell'applicazione

Per procedere con uno sviluppo controllato dei documenti e del codice si è scelto di adottare il sistema di ticketing **Redmine**.

La scelta di tale software è descritta nella sezione 9.1.1.1.

In questa sezione vi saranno molti riferimenti impliciti a contenuti del *Piano di Progetto v1.2.0* e del *Piano di Qualifica v1.2.0*.

### 8.1 Creare un nuovo progetto

La creazione di un progetto è compito del *Responsabile di Progetto*.

Un nuovo progetto rappresenta una macro-attività caratterizzata da molte sotto-attività supervisionate da un responsabile.

Per creare un nuovo progetto:

- Aprire **Progetti**;
- Selezionare **Nuovo progetto**;
- Assegnare un **Nome** breve ma significativo;
- Nel caso in cui si voglia creare un sotto-progetto indicare il nome del progetto padre dall'omonimo campo;
- **Identificativo**: scrivere in minuscolo ed indicare codice della fase a cui si riferisce (es. ndp-rr);
- Lasciare inalterati gli altri campi.

### 8.2 Creazione ticket

I ticket vengono creati da:

- **Responsabile di Progetto**: crea i ticket più importanti che rappresentano le macro fasi evidenziate dalla pianificazione (vedi *Piano di Progetto v1.2.0*);
- **Responsabile di Sotto-progetto**: crea i ticket per i processi non pianificati inizialmente, che si evidenziano necessari per l'avanzamento del sotto-progetto assegnato;
- **Verificatore**: crea i ticket per segnalare errori ed imprecisioni trovate durante il processo di verifica.

I ticket possono essere di tre tipologie:

- **Ticket di pianificazione**: rappresentano le macro-attività di maggiore importanza. Sono organizzate in una gerarchia con vari livelli di importanza. Tali attività vengono create da:
  - *Responsabile di Progetto* che durante la pianificazione identifica le attività più importanti e generali;
  - *Responsabile di Sotto-progetto* che durante lo svolgimento delle attività può scomporre in sotto-problemi l'attività indicata dal *Responsabile di Progetto*.



- **Ticket di realizzazione e controllo:** tutti i documenti redatti, durante la stesura attraversano due stadi:
  - **Realizzazione:** un redattore del documento effettua una prima stesura;
  - **Controllo:** un redattore, diverso da quello della precedente fase, esegue un primo controllo sui contenuti della parte scritta.
- **Ticket di verifica:** rappresentano gli errori identificati dai *Verificatori* durante il controllo che la realizzazione dell'attività sia conforme a quanto richiesto e che rispetti tutte le norme.

### 8.2.1 Ticket di pianificazione

- Selezionare **Nuova segnalazione** da menù principale;
- **Tracker:** indicare la natura del ticket:
  - **Documento:** stesura di un documento. Il tipo di attività svolta dal redattore del documento viene definito durante la rendicontazione;
  - **Codifica:** stesura di codice;
  - **Verifica:** macro-attività di verifica sul prodotto dei sotto-processi.
- **Oggetto:** descrizione breve e significativa;
- **Descrizione:** descrizione comprensibile e con riferimenti esterni mediante link se necessario;
- **Stato:** Plan;
- **Attività principale:** se si vuole creare una **sotto-attività** indicare l'id del ticket padre;
- **Categoria:** PDCA, solo se il ticket viene creato dal *Responsabile di Progetto*;
- **Assegnato a:** inserire il nome del responsabile;
- **Osservatori:** aggiungere eventuali collaboratori.

### 8.2.2 Ticket di realizzazione e controllo

- Selezionare **Nuova segnalazione** da menù principale;
- **Tracker:** indicare la natura del ticket:
  - **Documento:** stesura di un documento. Il tipo di attività svolta dal redattore del documento viene definito durante la rendicontazione;
  - **Codifica:** stesura di codice;
  - **Verifica:** attività di verifica sui prodotti dei processi.
- **Oggetto:** descrizione breve e significativa secondo il principio: nome ticket padre - attività da svolgere (realizzazione o controllo);
- **Descrizione:** descrizione comprensibile e con riferimenti esterni mediante link se necessario;



- **Stato:** New;
- **Attività principale:** se si vuole creare una **sotto-attività** indicare l'id del ticket padre;
- **Inizio:** dare una data di inizio presunta;
- **Scadenza:** dare una data di fine presunta;
- **Assegnato a:** inserire il nome del responsabile;
- **Osservatori:** aggiungere eventuali collaboratori.

### 8.2.3 Ticket di verifica

Un *Verificatore* per creare un *ticket di verifica* deve:

1. assicurarsi che esista all'interno del progetto l'attività *Verifica*.  
Su tale attività vi devono essere due sotto-attività: "Verifica - realizzazione", "Verifica - approvazione".  
Tutti i ticket creati devono essere sotto-attività di: "Verifica - realizzazione";
2. Creare quindi il ticket secondo le seguenti direttive:
  - Selezionare **Nuova segnalazione** da menù principale;
  - **Tracker:** Bug;
  - **Oggetto:** descrizione breve e significativa dell'errore incontrato;
  - **Descrizione:** descrivere in modo dettagliato e chiaro: la natura e la posizione dell'errore;
  - **Stato:** New;
  - **Attività principale:** tutti i ticket devono essere figli del ticket "Verifica - realizzazione" del progetto su cui si sta eseguendo la verifica;
  - **Assegnato a:** inserire il nome del responsabile del progetto padre (es. responsabile delle *Norme di Progetto*).

Tutti i campi non segnalati sono da lasciare vuoti.

Sarà poi compito del responsabile del progetto padre decidere a chi assegnare la correzione dell'errore. Nel caso in cui l'errore segnalato non sia considerato valido dal *Responsabile del sotto-progetto* verrà confermato il rifiuto dal *Responsabile di Progetto*.

### 8.2.4 Dipendenze temporali

Dopo la creazione del ticket, per aggiungere **dipendenze temporali** tra i ticket:

- Andare su **segnalazioni**;
- Aprire il link alla segnalazione a cui aggiungere la dipendenza;
- Nella sezione **segnalazioni correlate** premere **aggiungi**;
- Scegliere **segue** e indicare il numero della segnalazione che lo blocca ed eventuali giorni di slack.

Tutti i campi non segnalati sono da lasciare vuoti.



### 8.3 Aggiornamento ticket

Esistendo due tipologie di ticket, viene qui definito la procedura per effettuare l'aggiornamento di entrambe.

#### 8.3.1 Ticket di pianificazione

- Andare sul menù **Segnalazioni**;
- Selezionare il ticket di interesse;
- Cliccare il link **Aggiorna**;
- Commentare ciò che si è fatto sulla form **Note**;
- Cambiare lo stato del ticket secondo la seguente logica:
  - **Do**: quando un ticket è in questo stato indica che una o più persone stanno lavorando su tale attività;
  - **Check**: quando un ticket è in questo stato indica che una o più persone stanno lavorando sulla verifica di tale attività;
  - **Act**: l'attività è stata conclusa e verificata, e ne sono state tratte le conclusioni adeguate.
- Se viene concluso, aggiornare lo stato del ticket di pianificazione padre.

#### 8.3.2 Ticket di realizzazione e controllo

- Andare sul menù **Segnalazioni**;
- Selezionare il ticket di interesse;
- Cliccare il link **Aggiorna**;
- Indicare il tempo impiegato in ore;
- Indicare il tipo di attività svolta;
- Commentare ciò che si è fatto sulla form **Note**;
- Cambiare lo stato del ticket secondo la seguente logica:
  - **In Progress**: quando un ticket è in questo stato indica che una o più persone stanno lavorando su tale attività. La percentuale di completamento deve essere impostata tra lo 0% ed il 90%;
  - **Closed**: l'attività è stata conclusa. La percentuale di completamento dell'attività è al 100%.
- Aggiornare lo stato del ticket di pianificazione padre secondo tali principi:
  - ticket figlio passa da New a In Progress: il ticket padre passa da Plan a Do, o da Do a Check;
  - ticket figlio passa a Closed: il ticket padre deve essere in Do o Check;
  - tutti i ticket figli vengono chiusi: il ticket padre passa ad Act.





### 8.3.3 Ticket di verifica

- Andare sul menù **Segnalazioni**;
- Selezionare il ticket di interesse;
- Cliccare il link **Aggiorna**;
- Indicare il tempo impiegato in ore;
- Indicare Verifica come tipo di attività svolta;
- Commentare le correzione nella form **Note**;
- Cambiare lo stato del ticket secondo la seguente logica:
  - **In Progress**: quando un ticket è in questo stato indica che una o più persone stanno lavorando su tale attività. La percentuale di completamento deve essere impostata tra lo 0% ed il 90%;
  - **Closed**: l'attività è stata conclusa. La percentuale di completamento dell'attività è al 100%;
  - **Rejected**: l'attività di verifica è stata rifiutata dal *Responsabile del sotto-progetto* in accordo con il *Responsabile di Progetto*.
- Aggiornare lo stato del ticket di pianificazione padre secondo tali principi:
  - ticket figlio passa da New a In Progress: il ticket padre passa da Plan a Do, o da Do a Check;
  - ticket figlio passa a Closed: il ticket padre deve essere in Do o Check;
  - tutti i ticket figli vengono chiusi: il ticket padre passa ad Act.

## 8.4 Consigli di utilizzo

### 8.4.1 Pagina personale

Per avere una immediata visualizzazione dei ticket assegnati, è consigliato personalizzare la pagina personale:

- Andare alla **Pagina personale**;
- Cliccare il link **Personalizza la pagina**;
- Dal menù a tendina **La mia pagina di blocco**, selezionare **Le mie segnalazioni** e premere il pulsante verde +;
- Ripetere il punto precedente per aggiungere **Segnalazioni osservate**.

### 8.4.2 Visualizzare segnalazioni

Per avere una visualizzazione più chiara delle segnalazioni si consiglia di ordinarle per oggetto. Tale risultato può essere ottenuto premendo **Oggetto** dalla pagina **Segnalazioni**.



## 9 Ambiente di lavoro

### 9.1 Coordinamento

Il coordinamento del gruppo avviene tramite:

- Server dedicato con repository<sub>G</sub> Git<sub>G</sub>;
- Dropbox<sub>G</sub>;
- Google Drive<sub>G</sub> e Calendar.

#### 9.1.1 Server dedicato

Sul server dedicato sono installate alcune applicazioni web che facilitano la gestione del progetto.

Il server dedicato è raggiungibile via **SSH<sub>G</sub>** dai membri del gruppo tramite scambio di chiave pubblica/privata.

##### 9.1.1.1 Gestione del progetto

Come piattaforma di gestione del progetto è stato scelto **Redmine**. Redmine fornisce:

- Un sistema flessibile di gestione dei ticket;
- Il grafico Gantt<sub>G</sub> delle attività;
- Un calendario per organizzare i compiti;
- La visualizzazione del repository<sub>G</sub> associato al progetto;
- Un sistema di rendicontazione del tempo.

Sono state valutate diverse alternative ma, dopo un'attenta fase di test, nessuna di queste è stata ritenuta all'altezza di Redmine per quantità e qualità delle caratteristiche proposte.

Sono stati provati i seguenti software:

- **Gitlab/Gitorious:** entrambe le piattaforme propongono una gestione del progetto minimale e inadeguata alle esigenze del gruppo;
- **ProjectPier:** la gestione dei ticket non è stata ritenuta all'altezza del sistema di ticketing di Redmine;
- **TRAC:** raggiunge la completezza di Redmine in quanto a caratteristiche tramite l'installazione di plugin non ufficiali ma non fornisce un'interfaccia altrettanto immediata, aumentando il tempo speso dal team nella gestione del progetto.

#### 9.1.1.2 Versionamento

Sono state prese in considerazione diversi software di versionamento (Git<sub>G</sub>, SVN, Mercurial) prima di decidere di usare **Git<sub>G</sub>**. I motivi principali della scelta sono:

- **Flessibilità:** Git<sub>G</sub> è un repository<sub>G</sub> distribuito con la possibilità di *commit* e *revert* locali;



- **Esperienza del team:**  $\text{Git}_G$  è già stato usato da alcuni componenti del gruppo Don't Panic.

Sono stati creati due repository  $\text{Git}_G$ :

- **documents.git<sub>G</sub>:** disponibile all'indirizzo

`ssh://sw-eng@ntp.myopenproject.it:documents.git`

conterrà i sorgenti  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$  e gli script necessari alla stesura dei documenti;

- **src.git<sub>G</sub>:** disponibile all'indirizzo

`ssh://sw-eng@ntp.myopenproject.it:src.git`

conterrà i sorgenti dell'applicazione.

Il repository  $\text{Git}_G$  **documents.git** è suddiviso secondo la seguente struttura:

documents

- AnalisiDeiRequisiti
- Glossario
- LetteraDiPresentazione
- NormeDiProgetto
- PianoDiProgetto
- PianoDiQualifica
- StudioDiFattibilita
- Verbali
- template

Una volta terminata la fase di lavorazione di un documento, verrà creato un branch di verifica. In questo modo i *Verificatori* potranno lavorare parallelamente al resto del gruppo ed effettuare il merge delle loro modifiche, una volta terminato il lavoro di verifica.

Per ogni revisione verrà creato un tag così da identificare chiaramente la versione dei documenti presentati ad ogni revisione.

Eventuali suddivisioni del repository  $\text{Git}_G$  **src.git** avverranno non prima della fase di **Progettazione Architettuale**.

L'accesso al repository  $\text{Git}_G$  avverrà solo tramite connessioni sicure  $\text{SSH}_G$  con scambio di chiave pubblica/privata con il server (legame tra PC e server) all'indirizzo.

### 9.1.1.3 Integrazione Continua

Si è scelto di adottare **Jenkins** per applicare l'integrazione continua allo sviluppo del progetto.

Tale software permette di pianificare ed impostare la compilazione del codice. Mette inoltre a disposizione un cruscotto su cui è possibile visualizzare lo stato del codice prodotto. Tale software è infatti in grado di interagire con il software di versionamento, e se disponibile con software per l'esecuzione di test sul codice prodotto.

Attualmente Jenkins viene utilizzato per la compilazione dei documenti  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ . Tale software è disponibile alla pagina:

<http://ntp.myopenproject.it:8070/>



### 9.1.2 Dropbox

In questo repository<sub>G</sub> dovranno essere messi solo file che non necessitano di controllo di versione<sub>G</sub> e che sono comunque recuperabili da altre fonti (es. internet).

Principalmente in questo repository<sub>G</sub> verranno messi i file di installazione dei software utilizzati dal gruppo, specialmente quelli installabili nei propri PC.

Questa dovrebbe garantire l'uso dello stesso software per ogni componente del gruppo anche a livello di versione.

I manuali (software, libri, PDF<sub>G</sub>) di consultazione sono disponibili così da avere uniformità di versione e di informazione.

Altresì è presente nel repository<sub>G</sub> una cartella dove il server farà backup del database.

La possibilità di installare Dropbox<sub>G</sub> sul proprio PC dà a ogni componente del gruppo la possibilità di avere a disposizione documentazione e software anche offline.

### 9.1.3 Google Drive e Calendar

In questo repository<sub>G</sub> verranno messi i documenti che:

- Non necessitano di controllo di versione<sub>G</sub>;
- Hanno bisogno di grande interattività tra i componenti del gruppo;
- Possono essere acceduti tramite l'uso di un semplice browser.

Questo strumento dovrebbe permettere a 2 o più componenti del gruppo di interagire lavorando sugli stessi documenti contemporaneamente.

Google Drive<sub>G</sub> viene utilizzato come strumento di supporto allo sviluppo della documentazione e del software presente su Git<sub>G</sub>.

**Google Calendar** viene utilizzato all'interno del gruppo per gestire le risorse umane. In particolare con tale strumento viene utilizzato per notificare in quali giorni un determinato membro non può essere disponibile e per segnalare date rilevanti per il gruppo, come ad esempio le date delle riunioni.

## 9.2 Ambiente documentale

### 9.2.1 Pianificazione

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<sub>G</sub> ed è quindi eseguibile su ogni sistema operativo. È il successore ufficiale di OpenProj. Tale software è stato scelto in quanto possiede tali caratteristiche:

- Portabilità, essendo basato su Java<sub>G</sub>;
- Open-source;
- Genera grafici Gantt<sub>G</sub>;
- Genera automaticamente grafici Program Evaluation and Review Technique (PERT<sub>G</sub>), a partire dal Gantt<sub>G</sub>;
- Genera automaticamente la Work Breakdown Structure<sub>G</sub> (WBS<sub>G</sub>), a partire dal Gantt<sub>G</sub> con allocazione di risorse;



- Calcola i parametri  $\text{Schedule Variance}_G$  ( $\text{SV}_G$ ) e  $\text{Budget Variance}_G$  ( $\text{BV}_G$ );
- Compatibile con Microsoft Project 2010;
- Salvataggio su file  $\text{XML}_G$ : essendo un file testuale è possibile fare il merge dei file in caso di conflitti su  $\text{repository}_G$ .

## 9.2.2 Stesura documenti

### 9.2.2.1 $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$

Per la stesura dei documenti è stato scelto di utilizzare il sistema  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ . Il motivo principale dietro a questa scelta è la facilità di separazione tra contenuto e formattazione: con  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$  è possibile definire l'aspetto delle pagine in un file template condiviso da tutti i documenti. Altre soluzioni come Microsoft Office, LibreOffice o Google Docs non avrebbero consentito questa separazione, duplicando il lavoro di formattazione del testo e non garantendo un risultato uniforme.

Il grande numero di pacchetti esistenti consente di implementare funzionalità comuni in maniera semplice. L'estensibilità di  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$  può essere sfruttata per creare funzioni e variabili globali che rendono la scrittura del contenuto più corretta sotto un punto di vista semantico. Un esempio è dato dal comando `\role{ruolo}` che identifica ogni ruolo all'interno del progetto.

Per la scrittura di documenti  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$  l'editor consigliato è **TeXstudio**<sup>1</sup>.

### 9.2.2.2 Script

Per semplificare il lavoro di scrittura dei documenti sono stati creati alcuni script accessibili tramite il comando `make`. Lo script presente all'indirizzo `documents/Makefile` consente di:

- **Generare tutti i documenti:** eseguendo il comando `make all` tutti i documenti verranno generati;
- **Generare un archivio con tutti i documenti:** eseguendo il comando `make dist` tutti i documenti verranno generati e inseriti in un archivio;
- **Evidenziare le parole presenti nel Glossario:** eseguendo il comando `make glossary` tutti i file  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$  in `documents` verranno esaminati e tutte le parole presenti nel *Glossario v1.2.0* verranno evidenziate tramite un comando  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ . Tutte le parole evidenziate che non hanno una corrispondenza nel *Glossario v1.2.0* verranno riportate alla normalità.

Attraverso lo script presente in `documents/example/Makefile`, e condiviso da tutte le cartelle contenenti un documento, è possibile:

- **Generare il  $\text{pdf}_G$  del documento:** il comando `make build` dev'essere utilizzato al solo scopo di visualizzare un'anteprima durante il lavoro. Per la generazione del documento è stato scelto `Latexmk` per la sua possibilità di iterare i comandi `pdflatex`, `makeindex` e `makeglossaries` fino ad ottenere un documento con tutti i riferimenti risolti;

---

<sup>1</sup><http://texstudio.sourceforge.net/>



- **Eliminare i file generati da compilazioni errate:** il comando `make clean` consente di eliminare i file generati precedentemente tramite il comando `make build`, in particolare in seguito ad un errore di sintassi;
- **Calcolare l'indice Gulpease:** attraverso il comando `make gulpease` lo script estrarrà dal documento corrente il testo ed effettuerà il calcolo dell'indice di leggibilità Gulpease. L'indice è calcolato secondo la seguente formula:

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

- **Controllo ortografico:** attraverso il comando `make aspell` verrà invocato il programma Aspell per il documento corrente.

Si prevede la possibilità di estendere ulteriormente i Makefile presentati in base alle esigenze che si incontreranno nelle fasi di sviluppo.

### 9.2.2.3 Controllo ortografico

Il software per il controllo ortografico è **Aspell**<sup>2</sup>. Il programma è accessibile tramite il Makefile oppure tramite interfaccia grafica nell'editor consigliato TeXstudio.

### 9.2.2.4 Gestione Use Case e Requisiti

Per semplificare il tracciamento dei requisiti è stata realizzata internamente Tracy, un'applicazione web che consente di gestire requisiti, casi d'uso ed altre fonti. I membri del gruppo sono in grado di interagirci tramite browser per aggiungere, modificare ed eliminare elementi.

Tracy permette di esportare il codice  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$  necessario ad elencare requisiti, use case e le loro relazioni.

Per i casi d'uso genera un'anteprima indicativa del diagramma mostrando i casi d'uso del flusso di eventi principale. Tale diagramma non è conforme ad  $\text{UML}_G$ , il suo uso si limita a fornire un'idea visiva dei dati inseriti.

---

<sup>2</sup><http://aspell.net/>



Tracy

Home Logout (Matteo)

Home » Use Cases » 32

### View UseCase #32 (UC1.2.2)

Id Use Case	32
Title	Modifica della luce
Parent	UC1.2 - Modifica della scena
Description	L'utente può effettuare varie operazioni sulla luce selezionata, dall'effettuare una traslazione al modificarne le caratteristiche
Pre	Il sistema presenta una fonte di luce selezionata
Post	La fonte di luce selezionata è stata modificata

Operations

- List Sources
- Create ExternalSource
- Create Use Case
- Update UseCase
- Delete UseCase
- View Image

### Event Flow

#### Main Scenario

[Add new Event to Main Scenario](#)

Displaying 1-2 of 2 results.

Id Event	Category	Description	Refers To	Primary Actor	Order
45	Principale	Traslazione luce	UC1.2.2.1	Utente	1
46	Principale	Modifica caratteristiche luce	UC1.2.2.2	Utente	2

#### Alternate

[Add new Event to Alternate Scenario](#)

Id Event	Category	Description	Refers To	Primary Actor	Order
No results found.					

Figura 1: Strumento tracciamento requisiti - screenshot

### 9.2.2.5 Grafici UML

Per quanto concerne la modellazione dei diagrammi di caso d'uso è stato scelto l'editor **Visual Paradigm**<sup>3</sup>. Dopo aver valutato numerose alternative quali ArgoUML, Dia, LucidChart, StarUML, UMLet e Papyrus, Visual Paradigm è risultato l'unico software in grado di coniugare il completo supporto a UML<sub>G</sub> 2.0 con un'interfaccia semplice e responsiva.

Per la realizzazione dei diagrammi delle classi è stato scelto **WhiteStarUML**<sup>4</sup>. Questo software, a differenza di Visual Paradigm, fornisce una funzionalità di importazione inversa delle modifiche effettuate: una volta definite le classi ed aver generato il codice, è in grado di importare eventuali modifiche rilevate nel codice nel diagramma.

### 9.2.2.6 Documentazione

È stato scelto il sistema di documentazione **Doxygen**<sup>5</sup> che consente di inserire descrizioni di classi e metodi tramite commenti all'interno del codice. In questo modo la lettura del codice diventa un processo più immediato e la scrittura della documentazione avviene nello stesso file in cui viene scritto, diminuendo il rischio di inconsistenze tra codice e documentazione. Doxygen consente di generare la documentazione in formato HTML e in formato L<sup>A</sup>T<sub>E</sub>X.

## 9.3 Ambiente di sviluppo

Si è deciso di preparare una macchina virtuale per **Virtualbox**<sup>6</sup> contenente tutto il software necessario per sviluppare l'applicazione. Uniformando l'ambiente di sviluppo viene facilitata la comunicazione e la riproduzione delle segnalazioni.

<sup>3</sup><http://www.visual-paradigm.com/>

<sup>4</sup><http://sourceforge.net/projects/whitestaruml/>

<sup>5</sup><http://www.stack.nl/~dimitri/doxygen/>

<sup>6</sup><https://www.virtualbox.org/>



### 9.3.1 Framework

Per la realizzazione del progetto è stato scelto il framework<sub>G</sub> **Qt**<sub>G</sub><sup>7</sup>.

I vantaggi principali, rispetto a soluzioni alternative, sono:

- **Funzionalità 3D:** tra le numerose librerie disponibili sono presenti alcune funzionalità che semplificheranno lo sviluppo del software;
- **Qt<sub>G</sub> Designer:** esiste un editor grafico per disegnare l'interfaccia;
- **Qt<sub>G</sub> Linguist:** esiste un tool in grado di semplificare il supporto per più linguaggi;
- **Crossplatform:** Qt<sub>G</sub> è un framework<sub>G</sub> crossplatform e tra le piattaforme supportate ufficialmente sono presenti Windows<sub>G</sub>, OS X e molte distribuzioni GNU/Linux<sub>G</sub>;
- **Esperienza del team:** tutti i membri del team hanno già lavorato con Qt<sub>G</sub> durante l'insegnamento di Programmazione ad Oggetti;
- **Licenza GNU LGPL<sub>G</sub> v2.1:** Qt<sub>G</sub> è disponibile con licenza GNU LGPL<sub>G</sub> v2.1 e può essere quindi utilizzato nel nostro progetto senza che sia necessario rilasciare il codice prodotto.

QtCreator è l'IDE<sub>G</sub> che verrà utilizzato per lo sviluppo. È stato scelto, oltre che per l'ottima integrazione con Qt<sub>G</sub>, perché combina in un'unica interfaccia strumenti di editing, la documentazione ufficiale, strumenti di debugging, strumenti di profiling<sub>G</sub> e strumenti di versionamento. La libreria di riferimento attuale è la "Qt<sub>G</sub> libraries 4.8.4".

## 9.4 Ambiente di verifica

Vengono qui elencati e sommariamente descritti gli strumenti automatizzati per effettuare la verifica dei documenti redatti e del codice prodotto.

Le metriche ed i metodi per effettuare verifica sono ampiamente e dettagliatamente descritti nel *Piano di Qualifica v1.2.0*. A tale documento si fa inoltre riferimento per le caratteristiche di fondamentale importanza per la verifica degli strumenti qui riportati.

### 9.4.1 Documenti

Per la verifica dei documenti si utilizzano:

- **TeXstudio:** come segnalato nella sezione 9.2.2, per la scrittura di documenti è consigliato utilizzare l'ambiente grafico TeXstudio. Tale strumento integra i dizionari di OpenOffice.org e segnala i potenziali errori ortografici presenti nel testo durante la stesura del testo stesso;
- **Aspell:** strumento per la correzione tipografica dei documenti redatti in L<sup>A</sup>T<sub>E</sub>X. L'utilizzo di tale strumento è descritto nella sezione 9.2.2;
- **Tracy:** strumento realizzato dal gruppo Don't Panic, descritto nella sezione 9.2.2, permette di automatizzare il tracciamento dei requisiti;
- **Script:** i componenti del gruppo hanno scritto vari script per automatizzare il controllo dei documenti. Tali script, descritti nella sezione 9.2.2, automaticamente: calcolano l'indice Gulpease e marcano i termini presenti nel glossario del documento.

---

<sup>7</sup><http://qt-project.org/>





### 9.4.2 Codice

La verifica del codice è suddivisa in statica e dinamica.  
Per entrambe vengono riportati gli strumenti da utilizzare.

#### 9.4.2.1 Analisi Statica

- **CppCheck**<sup>8</sup>: analizzatore statico per individuare errori comuni del codice C++;
- **CCCC**<sup>9</sup> (C and C++ Code Counter): Misura metriche riguardanti codice sorgente in C++. Le più rilevanti sono:
  - **Complessità ciclomatica (McCabe's complexity)**: utilizzata per misurare la complessità di funzioni, moduli, metodi o classi di un programma;
  - **Information flow measure (IF4)**: misura il flusso di informazioni come suggerito da Henry e Kafura. Il valore è calcolato come  $(lunghezzafunzione)^2 \times fan-in \times fan-out$ ;
  - **Linee di codice per linee di commento (L\_C)**: rapporto tra linee di codice e linee di commento;
  - **Metodi per classe (WMC)**: numero pesato di metodi e funzioni per classe. Vi sono due indici per tale parametro: WMC1 misura il numero nominale di funzioni e metodi, WMCv conta il numero di funzioni e metodi accessibili da altri metodi e non conta quelli privati;
  - **Profondità albero gerarchie(DIT)**: la lunghezza del percorso più lungo che termina con il modulo in esame;
  - **Numero di figli (NOC)**: numero di moduli che ereditano direttamente dal modulo in esame;
  - **Accoppiamento tra gli oggetti(CBO)**: numero di moduli che sono accoppiati al modulo in esame come clienti o come fornitori di servizi.
- **Clang**<sup>10</sup>: compilatore che segnala errori e warning in modo molto espressivo;
- **GCC**<sup>11</sup>: compilatore che segnala errori e warning in modo dettagliato e chiaro.

#### 9.4.2.2 Analisi Dinamica

- **Valgrind**<sup>12</sup>: strumento open source per il debug di problemi di memoria, la ricerca dei memory leak<sub>G</sub> ed il profiling<sub>G</sub> del software;
- **Unit testing**: Per assicurare il corretto funzionamento del programma durante lo sviluppo dell'applicazione, verranno scritti dei test usando **QtTest**<sup>13</sup>. QtTest è un framework<sub>G</sub> per test di unità ed è un modulo di Qt<sub>G</sub> SDK.  
Per scrivere un test sarà necessario creare una classe specifica per il testing e utilizzare le macro QVERIFY e QCOMPARE definite da QtTest.

<sup>8</sup><http://cppcheck.sourceforge.net/>

<sup>9</sup><http://cccc.sourceforge.net/>

<sup>10</sup><http://clang.llvm.org/diagnostics.html>

<sup>11</sup><http://gcc.gnu.org/>

<sup>12</sup><http://valgrind.org/>

<sup>13</sup><http://qt-project.org/doc/qt-4.8/qttest.html>