

# CLIPS

Communication & Localization with Indoor Positioning Systems

---

UNIVERSITÀ DI PADOVA

NORME DI PROGETTO

*Oscar Elia Conti, Marco Zanella*

<b>Versione</b>	1.00
<b>Data Redazione</b>	24/12/2015
<b>Redazione</b>	
<b>Verifica</b>	
<b>Approvazione</b>	
<b>Uso</b>	
<b>Distribuzione</b>	

# Indice

<b>1</b>	<b>Introduzione</b>	<b>2</b>
1.1	Scopo del documento . . . . .	2
1.2	Scopo del progetto . . . . .	2
1.3	Glossario . . . . .	2
1.4	Riferimenti utili . . . . .	3
1.4.1	Riferimenti normativi . . . . .	3
1.4.2	Riferimenti informativi . . . . .	3
<b>2</b>	<b>Organizzazione interna</b>	<b>4</b>
2.1	Comunicazione . . . . .	4
2.1.1	Comunicazioni interne . . . . .	4
2.1.2	Comunicazioni esterne . . . . .	4
2.1.3	Composizione delle email . . . . .	4
2.1.3.1	Mittente . . . . .	4
2.1.3.2	Destinatario . . . . .	5
2.1.3.3	Oggetto . . . . .	5
2.1.3.4	Corpo . . . . .	5
2.1.3.5	Allegati . . . . .	5
2.2	Riunioni . . . . .	6
2.2.1	Riunioni interne . . . . .	6
2.2.1.1	Convocazione di una riunione interna . . . . .	6
2.2.1.2	Gestione di una riunioni interna . . . . .	7
2.2.2	Riunioni esterne . . . . .	7
2.2.2.1	Convocazione di una riunione esterna . . . . .	7
2.2.2.2	Gestione di una riunione esterna . . . . .	7
2.2.3	Verbale di una riunione . . . . .	7
2.3	Ruoli di progetto . . . . .	8
2.3.1	Responsabile di progetto . . . . .	8
2.3.2	Amministratore . . . . .	9
2.3.3	Analista . . . . .	10
2.3.4	Progettista . . . . .	10
2.3.5	Programmatore . . . . .	11
2.3.6	Verificatore . . . . .	11
2.3.7	Rotazione dei ruoli . . . . .	11
2.4	Strumenti . . . . .	11
2.4.1	Pianificazione . . . . .	11
2.4.2	Creazione dei grafici di Gantt . . . . .	12
2.4.3	Repository . . . . .	12
2.4.3.1	Utilizzo . . . . .	12

2.4.3.2	Struttura . . . . .	13
<b>3</b>	<b>Documentazione</b>	<b>14</b>
3.1	Struttura dei documenti . . . . .	14
3.1.1	Frontespizio . . . . .	14
3.1.2	Informazioni sul documento . . . . .	15
3.1.3	Diario delle modifiche . . . . .	15
3.1.4	Indice delle sezioni . . . . .	16
3.1.5	Indice delle tabelle . . . . .	16
3.1.6	Indice delle figure . . . . .	16
3.1.7	Introduzione . . . . .	16
3.1.8	Contenuto . . . . .	17
3.2	Norme tipografiche . . . . .	17
3.2.1	Formattazione generale . . . . .	17
3.2.1.1	Testatine . . . . .	17
3.2.1.2	Piè pagina . . . . .	17
3.2.1.3	Orfani e vedove . . . . .	17
3.2.2	Caratteri . . . . .	17
3.2.2.1	Virgolette . . . . .	17
3.2.2.2	Parentesi . . . . .	18
3.2.2.3	Punteggiatura . . . . .	18
3.2.2.4	Numeri . . . . .	18
3.2.2.5	Lettere . . . . .	18
3.2.3	Stile del testo . . . . .	18
3.2.3.1	Corsivo . . . . .	18
3.2.3.2	Grassetto . . . . .	18
3.2.3.3	Sottolineato . . . . .	19
3.2.3.4	Monospace . . . . .	19
3.2.3.5	Glossario . . . . .	19
3.2.4	Composizione del testo . . . . .	19
3.2.4.1	Elenchi . . . . .	19
3.2.4.2	Descrizioni . . . . .	19
3.2.4.3	Note a piè pagina . . . . .	20
3.2.5	Formati . . . . .	20
3.2.5.1	Date . . . . .	20
3.2.5.2	Orari . . . . .	20
3.2.5.3	URI . . . . .	20
3.2.5.4	Sigle . . . . .	21
3.2.5.5	Ruoli di progetto . . . . .	22
3.2.5.6	Fasi del progetto . . . . .	22
3.2.5.7	Revisioni . . . . .	22

3.2.5.8	Nomi	22
3.2.5.8.1	Nome del gruppo	22
3.2.5.8.2	Nome del progetto	22
3.2.5.8.3	Nome del proponente	22
3.2.5.8.4	Nome del committente	22
3.2.5.8.5	Nome proprio	22
3.2.5.8.6	Nome di un file	22
3.2.5.8.7	Nome di un documento	22
3.2.6	Componenti grafiche	22
3.2.6.1	Immagini	22
3.2.6.2	Tabelle	23
3.3	Tipologie di documenti	23
3.3.1	Documenti formali	23
3.3.2	Documenti informali	23
3.3.3	Glossario	24
3.3.4	Verbali	24
3.3.4.1	Verbali di riunioni interne	24
3.3.4.2	Verbali di riunioni esterne	24
3.4	Versionamento dei documenti	24
3.5	Avanzamento di un documento	24
3.6	Strumenti	24
3.6.1	Latex	24
3.6.1.1	Template	24
3.6.1.2	Comandi personalizzati	24
<b>4</b>	<b>Configurazione</b>	<b>25</b>
4.1	Controllo di versione	25
4.2	Richieste di modifica	25
4.2.1	Procedura di richiesta di modifica	25
4.3	Struttura del repository	26
4.4	Nomi dei file	27
4.5	Commit	27
4.6	Codifica dei file	28
4.7	Aggiornamento del repository	28
4.8	Visibilità del repository	28
<b>5</b>	<b>Sviluppo</b>	<b>29</b>
5.1	Studio di fattibilità	29
5.2	Analisi dei requisiti	29
5.2.1	Classificazione dei casi d'uso	29
5.2.2	Classificazione dei requisiti	30

5.2.3	Strumenti . . . . .	31
5.2.3.1	Software per la creazione di diagrammi UML . . . . .	31
<b>6</b>	<b>Verifica . . . . .</b>	<b>32</b>
6.1	Issue tracking . . . . .	32
6.1.1	Sintassi di una Issue . . . . .	32
6.2	Analisi ?? . . . . .	32
6.2.1	Analisi statica . . . . .	32
6.2.2	Analisi dinamica . . . . .	33
6.3	Verifica dei documenti . . . . .	33
6.3.1	Sintassi . . . . .	34
6.3.2	Periodi . . . . .	34
6.3.3	Struttura del documento . . . . .	34
6.4	Verifica dei diagrammi UML . . . . .	34
6.4.1	Diagrammi dei casi d'uso . . . . .	34
6.5	Strumenti . . . . .	35

# 1 Introduzione

## 1.1 Scopo del documento

Questo documento ha lo scopo di stabilire le norme che tutti i membri del gruppo *Leaf* dovranno rispettare durante lo sviluppo del progetto **CLIPS**, gli strumenti che dovranno utilizzare e le procedure che dovranno seguire.

Tutti i membri del gruppo sono tenuti a prendere visione di tale documento e rispettare le norme in esso contenute, al fine di garantire uniformità nello svolgimento del progetto, garantendo maggiore efficienza ed efficacia alle attività.

Il documento rivolge l'attenzione sui seguenti contenuti:

- organizzazione della comunicazione all'interno del gruppo e verso l'esterno;
- modalità di stesura dei documenti;
- descrizione delle metodologie di lavoro durante lo sviluppo del progetto;
- modalità di gestione e utilizzo del repository<sub>|G|</sub> ;
- strumenti per la gestione dell'ambiente di lavoro.

## 1.2 Scopo del progetto

Ancora da decidere

## 1.3 Glossario

Allo scopo di rendere più semplice e chiara la comprensione dei documenti viene allegato il *Glossario v1.00* nel quale verranno raccolte le spiegazioni di:

- terminologia tecnica;
- termini ambigui;
- abbreviazioni;
- acronimi.

Per evidenziare un termine presente in tale documento, esso verrà marcato con il pedice <sub>|G|</sub>.

## 1.4 Riferimenti utili

### 1.4.1 Riferimenti normativi

- Capitolati d'appalto:  
<http://www.math.unipd.it/~tullio/IS-1/2015/Progetto/Capitolati.html>

### 1.4.2 Riferimenti informativi

- Materiale di riferimento del corso di Ingegneria del Software:  
<http://www.math.unipd.it/~tullio/IS-1/2015/>;
- Portable Document Format:  
[http://en.wikipedia.org/wiki/Portable\\_Document\\_Format](http://en.wikipedia.org/wiki/Portable_Document_Format);
- Rappresentazione dei numeri:  
[https://en.wikipedia.org/wiki/ISO\\_31-0#Numbers](https://en.wikipedia.org/wiki/ISO_31-0#Numbers);
- Rappresentazione di date e orari:  
[https://en.wikipedia.org/wiki/ISO\\_8601](https://en.wikipedia.org/wiki/ISO_8601);
- Unicode:  
<http://en.wikipedia.org/wiki/Unicode>;
- Bluetooth low energy:  
[https://en.wikipedia.org/wiki/Bluetooth\\_low\\_energy](https://en.wikipedia.org/wiki/Bluetooth_low_energy);

## 2 Organizzazione interna

### 2.1 Comunicazione

#### 2.1.1 Comunicazioni interne

Per le comunicazioni interne verrà utilizzata una mailing list appositamente creata: [nome@mailing.list](mailto:nome@mailing.list).

Il componente del gruppo che vuole inviare una email a tutti gli altri componenti deve inviare il messaggio dalla sua casella di posta elettronica all'indirizzo [nome@mailing.list](mailto:nome@mailing.list). In questo modo l'email verrà spedita a tutti i membri del team. Tale sistema deve essere utilizzato solo per questioni riguardanti il progetto.

Per facilitare la comunicazione il gruppo può avvalersi degli strumenti di messaggistica istantanea quali Slack e Telegram. Inoltre, per le videochiamate, è possibile utilizzare di Skype.

Quando i sistemi sopra elencati venissero utilizzati per decisioni rilevanti per lo sviluppo del progetto è necessario stilare un verbale.

I membri del gruppo sono tenuti a prestare attenzione al numero di messaggi diffusi per non creare difficoltà di comunicazione.

#### 2.1.2 Comunicazioni esterne

Per la comunicazioni esterne è stato creato un apposito indirizzo di posta elettronica: [leaf.gruppo@gmail.com](mailto:leaf.gruppo@gmail.com).

Il *Responsabile di progetto* ha l'incarico di mantenere i contatti tra il team e le componenti esterne utilizzando tale indirizzo di posta elettronica. Inoltre è suo compito informare i membri del gruppo delle discussioni avvenute con componenti esterne: questo può essere fatto riassumendo la conversazione in una email e inviandola alla mailing list del gruppo.

#### 2.1.3 Composizione delle email

Questa sezione tratta le norme da rispettare nella composizione delle email, sia per la comunicazione interna che esterna.

**2.1.3.1 Mittente** Nel caso di comunicazione interna il mittente dovrà essere l'indirizzo di posta elettronica personale del membro del gruppo che ha scritto l'email, mentre in caso di comunicazione esterna l'unico indirizzo che deve essere utilizzato è [leaf.gruppo@gmail.com](mailto:leaf.gruppo@gmail.com).

Nel caso in cui la comunicazione debba avvenire tra un gruppo ristretto



di persone all'interno del gruppo, questi potranno utilizzare i loro indirizzi personali.

**2.1.3.2 Destinatario** Nel caso di comunicazione interna al gruppo l'unico destinatario deve essere [nome@mailing.list](mailto:nome@mailing.list) per facilitare l'invio a tutti i membri del gruppo stesso, mentre in caso di comunicazione esterna il destinatari possono essere il Prof. Tullio Vardanega, il Prof. Riccardo Cardin oppure i committenti del progetto, a seconda dello scopo dell'email.

**2.1.3.3 Oggetto** L'oggetto dell'email deve sintetizzare il contenuto dell'email stessa in modo più chiaro ed esaustivo possibile. Possibilmente l'oggetto di una nuova email deve essere differente rispetto alle email inviate e ricevute in precedenza, in modo tale da rendere facilmente identificabile ogni messaggio.

Per comporre un messaggio di risposta è necessario anteporre all'oggetto il prefisso "Re:", mentre nel caso di inoltrare di un messaggio è obbligatorio aggiungere il prefisso "I:". In entrambi i casi, le rimanenti parti dell'email non vanno modificate.

**2.1.3.4 Corpo** Il corpo del messaggio deve essere esaustivo, sintetico e deve essere comprensibile a tutti i destinatari del messaggio.

In caso di risposta o inoltrare è preferibile aggiungere la nuova parte di testo all'inizio dell'email per permettere una più facile lettura del contenuto.

Nell'eventualità che nel contenuto di un'email ci si debba riferire a persone è preferibile utilizzare la sintassi: "Nome Cognome", nel caso invece in cui si debba fare esplicito riferimento ad un ruolo di progetto è consigliabile riportarne per intero il nome.

**2.1.3.5 Allegati** È consigliato di non fare uso di allegati per lo scambio di documenti o file, a meno che non siano strettamente necessario. È preferibile, invece, caricare questi file in una cartella di Google Drive e inviare per email il link al documento o file desiderato.

Sia in caso di invio di allegati, che di link a documenti o file è buona norma inserire una breve descrizione di presentazione dell'allegato in modo tale che sia possibile in modo semplice capire di cosa si tratta.

Infine si chiede di prestare attenzione al formato dei documenti e file.

## 2.2 Riunioni

Il *Responsabile di progetto* ha il compito di indire le riunioni sia interne che esterne. Per ogni riunione il *Responsabile di progetto* dovrà inviare un'email di convocazione strutturata in questo modo:

- **oggetto:** convocazione della riunione N per il giorno AAAA-mm-GG
- **corpo:**
  - Data;
  - Ora;
  - Luogo;
  - Tipo di riunione;
  - Ordine del giorno.

Dove N rappresenta il numero della riunione e tipo indica se la riunione sia interna richiesta dal *Responsabile di progetto*, interna richiesta da uno o più dai membri del gruppo oppure esterna.

Le informazioni sulle riunioni devono essere presentate con più preavviso possibile, almeno tre giorni prima in modo tale che i membri del gruppo possano organizzare i loro impegni in modo tale da essere presenti alla riunione.

### 2.2.1 Riunioni interne

**2.2.1.1 Convocazione di una riunione interna** In generale, il compito di convocare le riunioni interne spetta al *Responsabile di progetto*, che può indire le riunioni quando più lo ritiene opportuno. Gli altri componenti del gruppo possono richiedere una riunione interna straordinaria, presentando al *Responsabile di progetto* le motivazioni per le quali si ritiene necessaria una riunione. In questi casi il *Responsabile di progetto* può:

- Autorizzare lo svolgimento della riunione;
- Negare lo svolgimento della riunione, nel caso in cui non ritenga le motivazioni presentate valide abbastanza da richiedere una riunione;
- Suggestire mezzi di comunicazione differenti.

In ogni caso spetta al *Responsabile di progetto* decidere data, ora e luogo dell'incontro contattando i membri del team e chiedendo loro la disponibilità. Questi sono tenuti a rispondere tempestivamente in modo tale da dare la possibilità al *Responsabile di progetto* di anticipare o posticipare la data o l'ora della riunione.

**2.2.1.2 Gestione di una riunioni interna** All'inizio di ogni riunione interna il *Responsabile di progetto* nomina un Segretario che ha il compito di redigere la minuta della riunione, catturando possibilmente tutti i soli gli aspetti più importanti della riunione stessa. Terminato l'incontro, il Segretario ha il compito di redigere il verbale dell'incontro. Questo verbale verrà archiviato nel repository del gruppo, per la consultazione de parte di tutti i membri.

Durante le riunioni i partecipanti devono tenere un comportamento che favorisca la discussione all'interno del gruppo e di tutti gli argomenti previsti.

## **2.2.2 Riunioni esterne**

**2.2.2.1 Convocazione di una riunione esterna** Il *Responsabile di progetto* ha il compito di fissare le riunioni esterne con i Proponenti o con i Committenti, contattandoli tramite la casella di posta elettronica del gruppo.

Il *Responsabile di progetto* ha, inoltre, il compito di accordarsi con i Proponenti o Committenti riguardo data, orario e luogo dell'incontro, tenendo conto anche della disponibilità degli elementi del gruppo e mettendo il condizione, per quanto possibile, di far partecipare tutti alla riunione.

**2.2.2.2 Gestione di una riunione esterna** Ad ogni riunione esterna il *Responsabile di progetto* deve chiedere la disponibilità ai Proponenti o Committenti di registrare l'incontro, in modo tale da redigere un verbale alla fine della riunione che sia quanto più fedele possibile a ciò di cui si è discusso durante l'incontro, in modo tale che, anche se ci fossero stati dei membri, questi possano disporre di un documento che riporti i temi trattati in modo esaustivo. In questo caso, come per le riunioni interne, viene nominato un Segretario che ha il compito di riascoltare l'incontro e di scriverne un verbale. In caso non fosse possibile registrare l'incontro il *Responsabile di progetto* deve redigere il verbale della riunione esterna, avvalendosi dell'aiuto di tutti i membri del team presenti all'incontro, al fine di avere una trascrizione più fedele possibile dei contenuti.

## **2.2.3 Verbale di una riunione**

Il verbale di una riunione è un documento nel quale vengono riassunti gli argomenti trattati e nel quale vengono indicate eventuali decisioni prese nel corso della riunione stessa.

Il compito della stesura del verbale spetta o ad un Segretario, nominato all'inizio di una riunione interna oppure alla fine di una riunione esterna nella quale è stato dato il permesso di registrare l'incontro, oppure al *Responsabile*

*di progetto* con l'aiuto di tutto il gruppo, nel caso di riunione esterna senza la possibilità di registrare quanto è stato detto.

I verbali dovranno avere queste struttura:

- Dove e quando si è svolta la riunione;
- Quali membri hanno partecipato alla riunione e quali membri erano, invece, assenti;
- Il nome dell'eventuale Segretario oppure del *Responsabile di progetto* che ha redatto il verbale;
- Tipo di riunione;
- Ordine del giorno;
- Eventuali argomenti dell'ordine del giorno non trattati;
- Gli interventi più significativi in ordine temporale;
- Suggerimenti, proposte, decisioni emerse durante la riunione;
- Dubbi, problematiche emerse durante la riunione;
- Eventuali compiti assegnati a membri del gruppo, con indicazione di nome, cognome e ruolo della persona a cui è stato assegnato;
- Eventuali argomenti da trattare la prossima riunione.

## 2.3 Ruoli di progetto

Per lo sviluppo collaborativo del progetto, ai membri del gruppo saranno assegnati dei ruoli differenti che corrispondono a figure professionali. Ogni membro dovrà ricoprire ogni ruolo almeno una volta ed è necessario garantire che il ruolo di ciascun membro del gruppo non sia in conflitto con il ruolo che ha ricoperto in passato. È compito del *Verificatore* controllare che siano rispettate queste condizioni, in caso contrario dovrà avvertire il *Responsabile di progetto* che dovrà risolvere il problema.

### 2.3.1 Responsabile di progetto

Il *Responsabile di progetto* rappresenta il progetto presso il fornitore e presso il committente, accentrando su di sé la responsabilità di scelta e approvazione. Per questo motivo ha responsabilità su:

- Pianificazione, coordinamento e controllo delle attività;
- Gestione delle risorse;
- Analisi e gestione dei rischi;
- Approvazione dei documenti;
- Approvazione dell'offerta economica;
- Convocazione delle riunioni interne;
- Relazioni esterne;
- Assegnazione delle attività a persone.

Per questi motivi ha il compito di:

- Assicurarci che le attività di verifica e validazione siano svolte seguendo le *Norme di progetto*;
- Garantire il rispetto dei ruoli e dei compiti assegnati nel *Piano di progetto*;

### **2.3.2 Amministratore**

L'*Amministratore* è responsabile del controllo e della gestione dell'ambiente di lavoro. In particolare deve preoccuparsi di:

- Equipaggiare l'ambiente di lavoro con strumenti, procedure, infrastrutture e servizi a supporto dei processi che permettano di automatizzare il più possibile le attività o parti di esse;
- Garantire che l'ambiente di lavoro sia completo, dotato di tutti gli strumenti necessari al progetto, ordinato e aggiornato;
- Controllare le versioni e configurazioni del prodotto;
- Gestire la documentazione, controllarne la diffusione, la disponibilità e l'archiviazione;
- Fornire procedure e strumenti per il monitoraggio e segnalazione per il controllo qualità;
- Risolvere problemi legati alla gestione dei processi tramite l'adozione di strumenti adatti.

L'*Amministratore* redige inoltre le *Norme di progetto* dove viene spiegato l'utilizzo degli strumenti, e deve redigere la sezione del *Piano di qualifica* dove vengono descritti gli strumenti e i metodi atti alla verifica. L'*Amministratore* non compie scelte gestionali, ma tecnologiche concordate con il *Responsabile di progetto*.

### 2.3.3 Analista

L'*Analista* è il responsabile delle attività di analisi. Le mansioni che gli competono riguardano:

- Comprendere la natura e la complessità del problema tramite l'analisi dei bisogni e delle fonti;
- Classificare i requisiti;
- Stendere i diagrammi dei casi d'uso;
- Assegnare i requisiti a parti distinte del sistema;
- Assicurarsi che i requisiti trovati siano conformi alle richieste del committente.

L'*Analista* non si occupa di trovare una soluzione al problema, ma lo definisce redigendo lo *Studio di fattibilità* e l'*Analisi dei requisiti*. Partecipa alla definizione del *Piano di qualifica* in quanto conosce a fondo il problema e deve avere chiari i livelli di qualità richiesti, oltre alle procedure per ottenerli.

### 2.3.4 Progettista

Il *Progettista* è il responsabile delle attività di progettazione. I compiti a lui affidati comprendono:

- Produrre una soluzione attuabile e che sia comprensibile e soddisfacente per stakeholder;
- Effettuare scelte su aspetti progettuali che applichino al prodotto soluzioni note ed ottimizzate;
- Effettuare scelte su aspetti progettuali e tecnologici che rendano il prodotto facilmente manutenibile;
- Effettuare scelte su aspetti progettuali e tecnologici che rendano il prodotto realizzabile con costi e scadenze prefissate;

Il *Progettista* redige i documenti di *Specifica tecnica*, *Definizione di prodotto* e si occupa delle sezioni del *Piano di qualifica* relative alle metriche di verifica della programmazione.

### 2.3.5 Programmatore

Il *Programmatore* è responsabile delle attività di codifica e delle componenti di ausilio necessarie per l'esecuzione delle prove di verifica e validazione. In particolare ha i seguenti compiti:

- Implementare in maniera rigorosa le soluzioni descritte dal *Progettista*;
- Scrivere codice che sia documentato, manutenibile e che rispetti le metriche stabilite per la scrittura del codice;
- Realizzare i test per la verifica e la validazione del codice stesso.

Il *Programmatore*, infine, deve occuparsi di redigere il *Manuale utente*.

### 2.3.6 Verificatore

Il *Verificatore* è il responsabile delle attività di verifica. Le mansioni che gli competono sono:

- Garantire che l'attuazione delle attività sia conforme alle norme stabilite;
- Verificare che le attività svolte non abbiano introdotto errori;
- Controllare la conformità di ogni stadio del ciclo di vita del prodotto.

Il *Verificatore* deve redigere le sezioni del *Piano di qualifica* riguardanti l'esito e la completezza delle verifiche effettuate.

### 2.3.7 Rotazione dei ruoli

Chiedere a chi fa il piano di progetto

## 2.4 Strumenti

### 2.4.1 Pianificazione

Lo strumento scelto per la gestione delle attività di pianificazione di progetto è Teamwork. Le caratteristiche rilevanti di questo software sono le seguenti:

- Creazione di attività e sotto-attività assegnabili ad uno o più membri del progetto;
- Possibilità di creare dipendenze tra le attività;
- Possibilità di assegnare priorità differenti ad ogni attività;
- Creazione di milestones da impostare sul calendario;
- Creazione automatica di grafici e report, tra cui il grafico di Gantt;
- Dashboard che permette di aver un riepilogo dello stato di avanzamento del progetto, con segnalazione di eventuali ritardi;
- Strumento per la segnalazione dei rischi;
- Sistema di gestione delle notifiche per ogni attività svolta;
- Versione mobile;

Questo strumento, dopo essere stato valutato insieme ad altri strumenti quali Trello, Freedcampe Zoho, è risultato essere il più completo.

#### **2.4.2 Creazione dei grafici di Gantt**

#### **2.4.3 Repository**

Per la gestione e l'organizzazione dei file inerenti la documentazione e la codifica, è stato creato un repository. Il repository risiede su GitHub, il quale utilizza il sistema di versionamento Git.

**2.4.3.1 Utilizzo** Ciascun membro del gruppo ha libero accesso al repository. Le principali operazioni da svolgere durante una sessione di lavoro sono i seguenti:

- **Pull:** scarica la versione più aggiornata del progetto per permettere di lavorare offline;
- **Add:** permette di aggiungere uno o più file alla lista di file da aggiornare, creando una proposta di modifica;
- **Commit:** permette di validare le modifiche effettuate in precedenza;
- **Push:** pubblica le modifiche effettuate in locale nel repository remoto;



- **Merge:** permette di fondere un branch all'interno del repository scelto in modo tale da rendere disponibili le modifiche effettuate all'interno del repository stesso.

Ogni membro del team deve sempre controllare i file che aggiunge all'interno del repository, facendo particolare attenzione ai file nascosti. Ogni commit deve sempre contenere un messaggio descrittivo del lavoro svolto.

**2.4.3.2 Struttura** Il repository contiene due rami principali: master e development.

Nel ramo master deve essere presente sempre l'ultima versione del progetto.

Nel ramo development è dove avviene lo sviluppo di nuove funzionalità.

?? Controllare

## 3 Documentazione

In questa sezione sono indicati gli standard riguardanti la struttura e la stesura dei documenti prodotti.

### 3.1 Struttura dei documenti

Ogni documento è realizzato a partire da una struttura prestabilita che dovrà essere uguale per tutti i documenti ufficiali ad eccezione dei verbali:

1. frontespizio;
2. informazione sul documento;
3. diario delle modifiche;
4. indice delle sezioni;
5. indice delle tabelle;
6. indice delle figure;
7. introduzione;
8. contenuto.

L'ordine di ognuna delle sezioni è fissato. La numerazione delle prime pagine sarà quella romana, mentre dall'introduzione fino alla fine del documento quella araba.

#### 3.1.1 Frontespizio

Questa sezione deve trovarsi nella prima pagina di ogni documento e contiene:

1. informazioni sul gruppo:
  - a. nome;
  - b. logo;
  - c. mail.
2. informazioni sul progetto:
  - a. nome progetto;
  - b. nome azienda proponente.

3. informazioni sul documento:

- a. nome;
- b. versione.

### **3.1.2 Informazioni sul documento**

In questa sezione vengono indicate le principali informazioni riguardanti il documento quali:

- 1. versione;
- 2. data di redazione;
- 3. cognome e nome di coloro che hanno redatto il documento (in ordine alfabetico)
- 4. cognome e nome di coloro che hanno verificato il documento (in ordine alfabetico)
- 5. ambito d'uso del documento (interno oppure esterno)
- 6. cognome e nome di coloro ai quali è destinato il documento (in ordine alfabetico)

### **3.1.3 Diario delle modifiche**

Questa sezione descrive, attraverso l'utilizzo di una tabella, le modifiche che sono state apportate al documento. Ogni riga della tabella corrisponde ad una modifica effettuata al documento. La struttura della riga della tabella è la seguente:

- 1. versione del documento;
- 2. data della modifica;
- 3. cognome e nome dell'autore della modifica;
- 4. ruolo dell'autore della modifica nel momento in cui essa è avvenuta;
- 5. sommario delle modifiche apportate

Le righe della tabella sono ordinate a partire dalla data dell'ultima modifica effettuata, in ordine cronologico inverso.

#### **3.1.4 Indice delle sezioni**

L'indice delle sezioni contiene l'indice di tutti gli argomenti trattati all'interno del documento. La sua struttura è la seguente:

1. titolo dell'argomento trattato;
2. numero di pagina.

#### **3.1.5 Indice delle tabelle**

Questa sezione contiene l'indice delle tabelle. Per ogni tabella deve essere specificato:

1. titolo della tabella;
2. numero di pagina di riferimento.

Nel caso in cui non siano presenti tabelle all'interno del documento, è possibile omettere questa sezione.

#### **3.1.6 Indice delle figure**

In questa sezione sono riportate tutte le figure presenti all'interno del documento. Per ogni figura deve essere specificato:

- nome figura;
- pagina di riferimento della figura.

Nel caso in cui non siano presenti figure all'interno del documento, è possibile omettere questa sezione.

#### **3.1.7 Introduzione**

Questa sezione deve riportare le seguenti informazioni:

1. scopo del documento;
2. glossario;
3. riferimenti utili:
  - a. riferimenti normativi;
  - b. riferimenti informativi;

### 3.1.8 Contenuto

Questa sezione contiene il contenuto del documento. Anch'esso deve essere propriamente diviso in sezioni e sottosezioni.

## 3.2 Norme tipografiche

### 3.2.1 Formattazione generale

**3.2.1.1 Testatine** Ogni pagina di un documento, fatta eccezione per il frontespizio, deve contenere la testina. Essa è composta da:

- logo del gruppo, posizionato in alto a sinistra;
- nome del documento, posizionato in alto a destra.

**3.2.1.2 Piè pagina** Ogni pagina di un documento, deve contenere il piè pagina. Esso contiene:

- numero della pagina, posizionato al centro.

**3.2.1.3 Orfani e vedove** Si considerano vedova, la riga di un paragrafo che inizia alla fine di una pagina, mentre si considera orfana, la riga di un paragrafo che finisce all'inizio di una pagina. I documenti dovranno essere redatti cercando di evitare il più possibile queste due tipologie di righe poiché risultano poco gradevoli.

### 3.2.2 Caratteri

#### 3.2.2.1 Virgolette

- **Virgolette alte singole** ‘ ’ : devono essere utilizzate per racchiudere un singolo carattere;
- **Virgolette alte doppie** “ ” : devono essere utilizzate per racchiudere nomi di file o comandi. Possono essere utilizzate anche per descrivere parole il cui ordine è prestabilito;
- **Virgolette basse** « » : devono essere utilizzate per racchiudere citazioni.

Non sono ammessi ulteriori casi d'uso per le virgolette.

### 3.2.2.2 Parentesi

- **Tonde:** possono essere utilizzate per descrivere esempi, per fornire dei sinonimi oppure per dare delle precisazioni. Sono le uniche parentesi ammesse all'interno di una frase.
- **Quadre:** possono rappresentare uno standard ISO oppure uno stato relativo ad un ticket.

**3.2.2.3 Punteggiatura** La punteggiatura deve essere sempre utilizzata attentamente per cercare di rendere il discorso il più chiaro e coeso possibile. Non sono ammesse spaziature prima dell'utilizzo di un carattere di punteggiatura. L'utilizzo del punto è necessario per indicare la fine di un concetto e poter iniziare un altro.

**3.2.2.4 Numeri** I numeri all'interno dei documenti devono essere formattati seguendo lo standard [SI/ISO 31-0]. Esso prevede che la parte frazionaria sia separata da quella decimale utilizzando la virgola. I numeri la cui parte intera supera le tre cifre, devono essere scritti raggruppando in gruppi di tre le cifre di cui è composta la parte intera, partendo dalla cifra meno significativa e separandoli con uno spazio unificatore.

**3.2.2.5 Lettere** Cosa dovrei scrivere delle lettere???

### 3.2.3 Stile del testo

**3.2.3.1 Corsivo** Il corsivo va utilizzato per riportare le seguenti informazioni:

- nome di un documento;
- nome di un ruolo;
- percorsi di cartelle.

**3.2.3.2 Grassetto** Il grassetto va utilizzato per riportare le seguenti informazioni:

- titoli;
- parole su cui è utile focalizzare l'attenzione del lettore all'interno di un argomento;
- parole chiave all'interno di elenchi.

**3.2.3.3 Sottolineato** La sottolineatura è indicata qualora si voglia evidenziare l'importanza di una parola all'interno di una frase.

**3.2.3.4 Monospace** Lo stile monospace va applicato nel caso in cui si vogliano riportare all'interno di un documento comandi oppure parti di codice.

**3.2.3.5 Glossario** Questo stile va applicato per tutte le parole che hanno una corrispondenza all'interno del glossario. Ogni parola presente nel glossario deve essere seguita da un pedice contenente il carattere 'g' scritto in corsivo. Non si applica questa regola nei casi in cui la parola compaia all'interno di titoli, percorsi, nomi di cartelle, comandi o parti di codice.

## **3.2.4 Composizione del testo**

**3.2.4.1 Elenchi** Le norme che regolano un elenco sono le seguenti:

- la prima parola di un elenco deve essere maiuscola, fatta eccezione nel caso in cui l'elenco inizi con il carattere ':';
- ogni elemento dell'elenco, tranne l'ultimo, deve terminare con il carattere ';'. È fatta eccezione nel caso in cui l'elemento sia composto da più frasi, allora è permesso il '.'.
- l'ultimo elemento di un elenco deve sempre terminare con il carattere '.'.

È necessario usare elenchi numerato quando è l'ordine degli elementi è rilevante. Per gli elenchi numerati valgono le seguenti regole:

- nel primo livello si usano numeri interni a partire da uno;
- nel secondo livello si usano lettere dell'alfabeto a partire dalla 'a'.

Gli elenchi puntati servono per descrivere elementi di cui non è importante l'ordine espositivo. Essi seguono le seguenti regole:

- nel primo livello bisogna utilizzare cerchi neri pieni;
- nel secondo livello trattini neri.

**3.2.4.2 Descrizioni** Serve??

**3.2.4.3 Note a piè pagina** Le note a piè pagina seguono le seguenti regole:

- la loro numerazione è progressiva all'interno di tutto il documento;
- devono essere scritte una volta sola;
- il primo carattere di ogni nota deve essere maiuscolo. Fanno eccezione i casi in cui la parola sia un acronimo. In questo caso bisogna seguire le regole di formattazione di tale acronimo.

### **3.2.5 Formati**

**3.2.5.1 Date** La formattazione delle date segue lo standard [ISO 8601]. Tale standard prevede che una data sia scritta secondo il seguente formalismo: AAAA-MM-GG. Questa rappresentazione va letta nel seguente modo:

- AAAA: numero a quattro cifre che rappresenta l'anno;
- MM: numero a due cifre che rappresenta il mese;
- GG: numero a due cifre che rappresenta il giorno.

Nei casi in cui risulti possibile esprimere i mesi o giorni omettendo una cifra, è necessario anteporre uno zero davanti a tale cifra.

**3.2.5.2 Orari** La formattazione degli orari segue lo standard [ISO 8601]. Tale standard prevede che gli orari siano scritti secondo il seguente formalismo: hh:mm. Questa rappresentazione va letta nel seguente modo:

- hh: numero a due cifre che rappresenta il numero di ore trascorse dalla mezzanotte;
- mm: numero a due cifre che rappresenta i minuti.

Nei casi in cui risulti possibile esprimere le ore o i minuti omettendo una cifra, è necessario anteporre uno zero davanti a tale cifra.

**3.2.5.3 URI** Lo stile utilizzato per rappresentare un URI è il corsivo ed il testo deve essere di colore blu.



**3.2.5.4 Sigle** È possibile fare riferimento a ruoli, documenti e revisioni pianificate utilizzando le seguenti sigle:

- Rdp (*Responsabile di progetto*);
- Amm (*Amministratore*);
- Ana (*Analista*);
- Ptt (*Progettista*);
- Prm (*Programmatore*);
- Ver (*Verificatore*).
- ADR (*Analisi dei requisiti*);
- GLS (*Glossario*);
- NDP (*Norme di progetto*);
- PDP (*Piano di progetto*);
- PDQ (*Piano di qualifica*);
- SDF (*Studio di fattibilità*);
- SPT (*Specificazione tecnica*);
- RR (**Revisione dei requisiti**);
- RA (**Revisione di accettazione**);
- RP (**Revisione di progettazione**);
- RQ (**Revisione di qualifica**);

L'utilizzo di tale sigle è permesso solo all'interno di:

- Tabelle;
- Diagrammi (immagini);
- Didascalie di tabelle e immagini;
- Note a piè di pagina.

**3.2.5.5 Ruoli di progetto** Quando si fa riferimento ad un ruolo di progetto bisogna adottare lo stile corsivo e la prima lettera deve essere maiuscola.

**3.2.5.6 Fasi del progetto** Quando si fa riferimento ad una fase del progetto bisogna adottare lo stile grassetto e la prima lettera deve essere maiuscola.

**3.2.5.7 Revisioni** Quando si fa riferimento ad una revisione bisogna adottare lo stile grassetto e la prima lettera deve essere maiuscola.

#### **3.2.5.8 Nomi**

**3.2.5.8.1 Nome del gruppo**

**3.2.5.8.2 Nome del progetto**

**3.2.5.8.3 Nome del proponente**

**3.2.5.8.4 Nome del committente**

**3.2.5.8.5 Nome proprio** I nomi proprio vanno scritti ponendo il cognome davanti al nome. L'ordine da seguire nel caso di elenchi è quello alfabetico.

**3.2.5.8.6 Nome di un file** I nomi dei file vanno formattati utilizzando lo stile monospace e devono essere racchiusi dalle doppie virgolette alte.

**3.2.5.8.7 Nome di un documento** Quando si fa riferimento ad un documento bisogna adottare lo stile corsivo e la prima lettera deve essere maiuscola. I nomi dei documenti devono essere racchiusi dalle doppie virgolette alte.

#### **3.2.6 Componenti grafiche**

**3.2.6.1 Immagini** L'utilizzo delle immagini all'interno di un documento è regolamentato secondo quanto segue:

- i formati ammessi per le immagini sono il PNG e il PDF;

- devono essere numerate in ordine crescente;
- devono essere seguite da una breve descrizione;
- deve essere presente un riferimento all'immagine all'interno dell'indice immagini.

**3.2.6.2 Tabelle** L'utilizzo delle tabelle all'interno di un documento è regolamentato secondo quanto segue:

- devono essere numerate in ordine crescente;
- devono essere seguite da una breve descrizione;
- devono essere presente un riferimento alla tabella all'interno dell'indice delle tabelle.

Valutare colori tabelle?????!!!! Vedi ProTech.

### **3.3 Tipologie di documenti**

#### **3.3.1 Documenti formali**

I documenti formali possono essere descritti secondo quanto segue:

- sono documenti approvati dal responsabile di progetto;
- eventuali modifiche ad un documento formale, lo rendono informale;
- sono gli unici documenti che possono essere distribuiti all'esterno del team di progetto;

#### **3.3.2 Documenti informali**

I documenti informali possono essere descritti secondo quanto segue:

- sono documenti non ancora approvati dal responsabile di progetto;
- possono essere distribuiti solamente all'interno del team di progetto;
- possono essere sottoposti a revisione;

### **3.3.3 Glossario**

Il glossario nasce dall'esigenza di chiarire il significato ambiguo che possono avere certe parole all'interno di determinati contesti. Al suo interno saranno quindi presenti alcune parole, prese dai documenti, che hanno le seguenti caratteristiche:

- trattano argomenti tecnici;
- trattano argomenti poco conosciuti o che possono scatenare ambiguità;
- rappresentano delle sigle;

Il glossario deve essere strutturato secondo quanto segue:

- i termini devono seguire l'ordine lessicografico;
- ogni termine deve essere seguito da una spiegazione chiara e concisa del significato del termine stesso. Questa spiegazione non deve essere in alcun modo ambigua.

Per evitare confusione, la stesura del glossario deve avvenire in maniera parallela alla stesura dei documenti. Al fine di evitare dimenticanze, è ammesso inserire un termine all'interno del glossario senza inserirne immediatamente la spiegazione. È comunque doveroso completare la spiegazione non appena possibile.

### **3.3.4 Verbali**

#### **3.3.4.1 Verbali di riunioni interne**

#### **3.3.4.2 Verbali di riunioni esterne**

## **3.4 Versionamento dei documenti**

## **3.5 Avanzamento di un documento**

## **3.6 Strumenti**

### **3.6.1 Latex**

#### **3.6.1.1 Template**

#### **3.6.1.2 Comandi personalizzati**

## 4 Configurazione

### 4.1 Controllo di versione

Il controllo di versione di documenti e file sorgente viene fatto utilizzando il software Git. Nonostante siano presenti varie alternative, come Mercurial, Subversion e Bazaar, è stato scelto Git in quanto soddisfa appieno le necessità di versionamento dei file per questo progetto e, inoltre, permette di versionare i file localmente, senza bisogno di una connessione internet attiva. Come servizio di hosting per la repository è stato scelto GitHub. Entrambe le scelte sono state fatte anche perché più membri del gruppo avevano già avuto la possibilità di lavorare con tali servizi.

Ogni qualvolta viene eseguita una modifica sostanziale ad un documento o ad un file sorgente deve essere assegnato un nuovo numero di versione a questo, per tener traccia delle modifiche fatte.

Per quanto riguarda le norme di versionamento dei file inerenti alla documentazione si rimanda alla sezione [Versionamento dei documenti](#) del presente documento.

### 4.2 Richieste di modifica

Ogni componente del gruppo può avanzare una richiesta di modifica al *Responsabile di progetto*, qualora lo ritenga necessario. Il *Responsabile di progetto* ha il compito di analizzare tale richiesta e decidere se approvarla o meno. In caso affermativo, deve assegnare il compito di realizzare tale modifica ad un membro del gruppo, in base al ruolo da lui ricoperto in quel momento. Una volta effettuata la modifica questa deve essere sottoposta a verifica e dev'essere fatta mantenendo traccia dello stato precedente. In ogni caso, sia di accettazione della richiesta di modifica, sia in caso di rifiuto, il *Responsabile di progetto* è tenuto a motivare la scelta.

#### 4.2.1 Procedura di richiesta di modifica

Il membro del gruppo che vuole effettuare una modifica deve presentare una richiesta formale al *Responsabile di progetto*. La richiesta deve avere i seguenti campi:

1. Autore: contiene nome, cognome e ruolo del richiedente della modifica;
2. Documento: contiene il nome del documento di cui si ritiene in cui andrebbe fatta tale modifica;

3. Urgenza: indica una misura di quanto il richiedente ritiene necessario fare una modifica, può assumere solamente tali valori:
  - (a) “Alta”: in questo caso si ritiene che la modifica da fare sia importante, con pesanti conseguenza nell’organizzazione dello svolgimento delle attività future o che possa in qualche modo compromettere l’organizzazione delle attività future;
  - (b) “Media”: in questo caso il richiedente considera la modifica importante, ma non comporta forti conseguenza nell’organizzazione generale delle attività, ma può influenzare lo svolgimento di alcune di queste;
  - (c) “Bassa”: in questo caso la modifica proposta di modifica è di importanza secondaria, un suggerimento nell’effettuare alcune attività oppure che non influisce, o il suo peso è poco rilevante, nello svolgimento delle attività.
4. Descrizione: una descrizione dettagliata e motivata delle modifiche che si vogliono apportare;
5. Decisione del *Responsabile di progetto*. Questo campo va aggiunto successivamente alla decisione del *Responsabile di progetto*. Questo campo può assumere due valori:
  - (a) “Approvata”;
  - (b) “Respinta”.

Il *Responsabile di progetto* può aggiungere a questo campo la motivazione per cui ha approvato o meno la modifica, in caso lo ritenga necessario.

### 4.3 Struttura del repository

I file all’interno del repository verranno organizzati secondo questa struttura:

- /Documents
  - NormeDiProgetto;
  - StudioDiFattibilità;
  - AnalisiDeiRequisiti;
  - PianoDiProgetto;
  - PianoDiQualifica;

- Verbali;
- Glossario.
- /Source

La struttura di /Source verrà decisa all’inizio della progettazione architettuale.

## 4.4 Nomi dei file

I nomi dei file interni al repository devono sottostare alle seguenti norme:

- Devono contenere solo lettere, numeri, il carattere ‘\_’, il segno meno ed il punto;
- Devono avere lunghezza minima di tre caratteri;
- Devono identificare in modo non ambiguo i file;
- Devono riportare le informazioni dal generale al particolare;
- Devono, nel caso contengano date, rispettare il formato YYYY-MM-DD.

È consigliato invece:

- Utilizzare la notazione CamelCase, invece di caratteri ‘\_’ e ‘-’ qualora il nome di un file fosse composto da più parole;
- Utilizzare nomi né troppo lunghi, né troppo corti, indicativamente tra i 10 e i 25 caratteri, estensione compresa;
- Specificare, quando possibile, l’estensione del file.

## 4.5 Commit

L’esecuzione di ogni comando commit deve sottostare alle seguenti norme:

- Ad ogni commit è necessario specificare un messaggio nel quale si deve dare una descrizione sintetica e più possibile precisa delle modifiche effettuate;
- Le modifiche apportate devono essere complete e testate con successo;
- È vietato l’utilizzo del comando “git add \*” prima di un comando commit per evitare di includere nel repository file nascosti, temporanei o non voluti.

## 4.6 Codifica dei file

I file contenenti codice oppure documentazione dovranno utilizzare la codifica UTF-8 senza BOM.

## 4.7 Aggiornamento del repository

Per l'aggiornamento del repository è prevista la seguente procedura:

- Dare il comando “git pull”. Nel caso in cui si verifichino dei conflitti:
  - Dare il comando “git stash” per accantonare momentaneamente le modifiche apportate;
  - Dare il comando “git pull”;
  - Dare il comando “git stash apply” per ripristinare le modifiche.

In questo modo il repository risulta aggiornato rispetto il repository remoto;

- Dare il comando “git add NomeDelFile”, dove al posto di “NomeDelFile” si deve mettere il nome del file, o lista di nomi dei file, sul quale sono state effettuate delle modifiche;
- Dare il comando “git commit -m “Riassunto delle modifiche” ”, dove al posto di “Riassunto delle modifiche” deve essere specificato, tra virgolette, il riassunto delle modifiche effettuate;
- Dare il comando “git push”.

## 4.8 Visibilità del repository

Per la condivisione e il versionamento dei configuration item è stato creato un repository privato su GitHub, raggiungibile all'indirizzo <https://github.com/mzanello/Leaf>. L'accesso è consentito solo ai membri del gruppo.



## 5 Sviluppo

### 5.1 Studio di fattibilità

Il documento riguardante lo *Studio di fattibilità* deve essere redatto rapidamente ed in modo accurato dagli *Analisti* sulla base di ciò che è emerso nelle prime riunioni, nelle quali si deve discutere di temi riguardanti i capitolati, come:

- Rischi nell'affrontare ogni capitolato;
- Rapporto tra i costi ed i benefici, sia in base al mercato attuale che futuro, sia in base al costo di produzione e alla possibile redditività futura;
- Il dominio applicativo e tecnologico di ogni capitolato.

### 5.2 Analisi dei requisiti

L'*Analisi dei requisiti* è il documento dove devono essere catalogati e descritti tutti i requisiti che il prodotto finale deve soddisfare. Ogni requisito deve emergere da una delle seguenti fonti:

- Capitolati d'appalto;
- Incontri con il proponente;
- Incontri con il committente;
- Valutazioni effettuate durante riunioni interne al gruppo.

Tele documento deve inoltre riportare il modo in cui ogni requisito deve essere verificato.

#### 5.2.1 Classificazione dei casi d'uso

È compito degli *Analisti* redigere una descrizione, dare una classificazione e fornire un diagramma conforme allo standard UML per ogni caso d'uso. Ogni caso d'uso dev'essere descritto con le seguenti informazioni, possibilmente in quest'ordine:

1. Codice identificativo del caso d'uso, nella forma

UCXY

dove:

- **X** (da spiegare meglio quando messo giù cosa si fa nell'introduzione);
  - **Y** è un codice numerico gerarchico.
2. Titolo, che deve descrivere sinteticamente il caso d'uso;
  3. Attori principali;
  4. Attori secondari, se questi sono presenti;
  5. Precondizioni, ovvero le condizioni che necessariamente devono verificarsi prima del caso d'uso;
  6. Postcondizioni, ciò che deve essere verificato successivamente al caso d'uso;
  7. Flusso principale degli eventi, dove si descrive il flusso dei casi d'uso figli. Per ogni evento va specificato:
    - Una descrizione testuale dell'evento;
    - Gli attori coinvolti;
    - Se l'azione è descritta dettagliatamente da un altro caso d'uso.
  8. Scenari alternativi, ovvero scenari in cui si verificano eccezioni o errori. Per ognuno di questi deve essere indicato:
    - Una descrizione testuale dell'evento;
    - Gli attori coinvolti;
    - Se l'azione è descritta dettagliatamente da un altro caso d'uso.

### 5.2.2 Classificazione dei requisiti

È compito degli *Analisti* redigere e classificare i requisiti del prodotto. I requisiti devono essere classificati in base al tipo e alla priorità, utilizzando la seguente notazione:

$$R[X][Y][Z]$$

dove:

1. **X** indica l'importanza strategica del requisito. Deve assumere solo i seguenti valori:

- **Obb**: Indica un requisito obbligatorio;
  - **Des**: Indica un requisito desiderabile;
  - **Opz**: Indica un requisito opzionale.
2. **Y** indica la tipologia del requisito. Deve assumere solo i seguenti valori:
- **F**: Indica un requisito funzionale;
  - **Q**: Indica un requisito qualità;
  - **P**: Indica un requisito prestazionale;
  - **V**: Indica un requisito vincolo;
3. **Z** rappresenta il codice univoco di ogni requisito in forma gerarchica.

### **5.2.3 Strumenti**

#### **5.2.3.1 Software per la creazione di diagrammi UML**

## 6 Verifica

### 6.1 Issue tracking

L'issue tracking è un'attività di supporto per i *Verificatori* e permette, oltre a tracciare potenziali errori, di assegnare i compiti di correzione al *Responsabile di progetto*.

#### 6.1.1 Sintassi di una Issue

Ogni issue avrà un nome che dovrà seguire la seguente notazione:

$$[D]:[T]-[S]$$

dove:

- **D** rappresenta la sigla del documento di interesse;
- **T** rappresenta la tipologia di issue;
- **S** rappresenta la sintesi della descrizione del problema.

Segue, poi, la descrizione della issue. Questa dovrà obbligatoriamente contenere:

- Dove, nel documento, si trova il problema. Questo va specificato tramite sezione e, possibilmente, il numero di riga dove inizia il problema;
- Una descrizione dettagliata del problema;
- La motivazione per cui si è ritenuto necessario sollevare una issue.

### 6.2 Analisi ??

Al fine di determinare il livello di qualità del prodotto, il team deve utilizzare le tecniche di analisi statica e dinamica dei dati.

#### 6.2.1 Analisi statica

Per analisi statica si intende la valutazione di un sistema basata su struttura, contenuto e documentazione senza che questo venga eseguito. Questa tecnica è applicabile sia al codice, che alla documentazione stessa. L'analisi statica può avvenire con due modalità:

- **Walkthrough:** questa tecnica dev'essere applicata quando non si sa che errori o problematiche si stanno cercando. La tecnica consiste nel leggere il codice sorgente o il documento da cima in fondo per trovare anomalie di qualsiasi tipo;
- **Inspection:** questa tecnica dev'essere applicata quando si ha idea della problematica che si sta cercando e consiste in una lettura mirata del documento/codice, sulla base di una lista degli errori precedentemente stilata.

La tecnica walkthrough è molto onerosa e deve essere utilizzata soprattutto nelle prime fasi del progetto, quando non è già presente una lista degli errori comuni, oppure non si è ancora sufficientemente preparati riguardo un aspetto del progetto. Avanzando nelle fasi del progetto sarà utile stilare una lista quanto più possibile completa di errori comuni, in modo tale da evitare la tecnica walkthrough e applicare l'inspection.

### 6.2.2 Analisi dinamica

L'analisi dinamica è una forma di valutazione di un sistema software, oppure di qualche suo componente, basato sull'osservazione del suo comportamento durante l'esecuzione. Questa tecnica non è applicabile, quindi, per trovare errori nella documentazione. I test che devono essere implementati devono essere in numero relativamente ridotto e il più possibile di valore dimostrativo.

## 6.3 Verifica dei documenti

Il *Responsabile di progetto* ha il compito di dare inizio alla fase di verifica, assegnando i compiti ai *Verificatori*. Quest'ultimi, devono effettuare un'accurata verifica delle seguenti regole:

- Deve essere utilizzata una sintassi corretta e il più possibile semplice;
- Devono essere utilizzati periodi brevi;
- La struttura del documento deve essere semplice e intuitiva;
- Devono essere rispettate le [norme tipografiche](#);
- Devono essere rispettate le [regole riguardanti il glossario](#).

Durante l'intera attività di verifica i *Verificatori* dovranno utilizzare lo strumento del diario delle modifiche in modo tale da concentrare la loro attenzione sulle modifiche effettuate ad un documento e tenere traccia degli errori più comuni commessi nel redigere i documenti.

### 6.3.1 Sintassi

I *Verificatori* hanno il compito di identificare errori sintattici nei documenti. Ciò può essere fatto con l'ausilio di strumenti automatici, ma ogni documento deve essere sempre sottoposto a walkthrough in modo tale da individuare errori che non sono stati in grado di evidenziare. Inoltre hanno il compito di inserire nel *Glossario* tutte quelle parole che possono essere fonte di ambiguità.

### 6.3.2 Periodi

I *Verificatori* hanno il compito di calcolare l'indice Gulpease di ogni documento utilizzando lo strumento automatico preposto (vedi sezione [Strumenti](#)). Qualora non si ottenga un risultato soddisfacente il *Verificatore* deve applicare la tecnica walkthrough con l'obiettivo di individuare periodi troppo lunghi, che possono essere di difficile comprensione o leggibilità.

### 6.3.3 Struttura del documento

I *Verificatori* devono verificare che la struttura di ogni documento rispetti i contenuti del documento stesso.

## 6.4 Verifica dei diagrammi UML

I *Verificatori* devono controllare tutti i diagrammi UML prodotti, sia che venga rispettato lo standard UML, sia che siano corretti semanticamente. I diagrammi utilizzati finora sono solamente i diagrammi dei casi d'uso.

### 6.4.1 Diagrammi dei casi d'uso

Le verifiche che devono essere effettuate ai diagrammi dei casi d'uso, innanzitutto, devono riguardare il rispetto dello standard UML, soprattutto riguardanti inclusioni, generalizzazioni e estensioni. Successivamente si deve verificare che ciò che descrive il diagramma sia effettivamente il sistema che vogliamo rappresentare. Per quest'ultimo punto bisogna porre particolare attenzione sull'identificazione degli attori.

Per identificare un attore può risultare utile seguire la seguente procedura:

1. Per prima cosa è utile chiedersi se ciò che vogliamo rappresentare come attore è una persona che interagisce col sistema. In caso affermativo è giusto rappresentarlo come attore, in caso negativo si deve andare al passo 2;
2. Se ciò che vogliamo rappresentare come attore non è una persona che interagisce col sistema bisogna chiedersi se questa “cosa” può cambiare insieme al design del sistema. In caso affermativo probabilmente ciò che vogliamo rappresentare non è un attore ma una parte del sistema stesso. In caso negativo, invece, è con buona probabilità un attore.

## **6.5 Strumenti**

Da fare