

2 marzo 2015



# Norme di Progetto

Informazioni sul documento	
Nome Documento	Norme di Progetto
Versione	1.0
Stato	<i>Formale</i>
Uso	<i>Interno</i>
Data Creazione	2 marzo 2015
Data Ultima Modifica	2 marzo 2015
Redazione	Busetto Matteo, Tollot Pietro, Petrucci Mauro, Fossa Manuel
Approvazione	Tollot Pietro
Verifica	Venturelli Giovanni
Lista distribuzione	<i>LateButSafe</i> Prof. Tullio Vardanega Prof. Riccardo Cardin Proponente Zucchetti S.p.a.



# Sommario

Il presente documento contiene le norme e le convenzioni che il gruppo LateButSafe intende adottare durante l'intero ciclo di vita del prodotto software Premi.

## Registro delle modifiche

Tab 1: Versionamento del documento

Versione	Autore	Data	Descrizione
1.3.0	Petrucci Mauro	4 Maggio 2015	Inserimento diagrammi attività
1.2.0	Petrucci Mauro	3 Maggio 2015	Incremento contenuti
1.1.0	Petrucci Mauro	1 Maggio 2015	Riorganizzazione capitoli
1.0.0	Petrucci Mauro	13 aprile 2015	Approvazione del documento
0.8.0	Petrucci Mauro	10 aprile 2015	Apportate le modifiche segnalate dal verificatore Venturelli Giovanni
0.5.0	Busetto Matteo	3 aprile 2015	Aggiornamento contenuti
0.4.0	Fossa Manuel	30 marzo 2015	Aggiornamento contenuti
0.3.0	Petrucci Mauro	22 marzo 2015	Correzione della parte Ingegneria dei Requisiti
0.2.1	Busetto Matteo	19 marzo 2015	Correzione di errori grammaticali e ortografici
0.2.0	Busetto Matteo	10 marzo 2015	Inserimento dei contenuti
0.1.0	Busetto Matteo	2 marzo 2015	Stesura dello scheletro del documento



## pre-RR

Tab 2: Storico ruoli pre-RR



# Indice

<b>1</b>	<b>Introduzione</b>	<b>7</b>
1.1	Scopo del documento . . . . .	7
1.2	Glossario . . . . .	7
1.3	Riferimenti . . . . .	7
1.3.1	Informativi . . . . .	7
<b>2</b>	<b>Ruoli di progetto</b>	<b>8</b>
2.1	Responsabile di Progetto . . . . .	8
2.2	Amministratore . . . . .	8
2.3	Analista . . . . .	9
2.4	Progettista . . . . .	9
2.5	Verificatore . . . . .	9
2.6	Programmatore . . . . .	10
<b>3</b>	<b>Collaborazione</b>	<b>11</b>
3.1	Comunicazioni . . . . .	11
3.1.1	Comunicazioni interne . . . . .	11
3.1.2	Comunicazioni esterne . . . . .	11
3.2	Riunioni . . . . .	11
3.2.1	Interne . . . . .	11
3.2.1.1	Casi Particolari . . . . .	11
3.2.2	Esterne . . . . .	12
3.2.3	Esito . . . . .	12
3.3	Repository e strumenti per la condivisione di file . . . . .	12
3.3.1	Repository . . . . .	12
3.3.2	Condivisione file . . . . .	12
<b>4</b>	<b>Documentazione</b>	<b>13</b>
4.1	Template . . . . .	13
4.2	Contenuto e struttura dei documenti . . . . .	13
4.2.1	Verbali . . . . .	13
4.2.2	Lettera di presentazione . . . . .	14
4.3	Norme tipografiche . . . . .	14
4.4	Formati di riferimento e altro . . . . .	15
4.5	Immagini e tabelle . . . . .	16
4.6	Glossario . . . . .	16
4.6.1	Implementazione . . . . .	17
<b>5</b>	<b>Procedure al supporto dei processi</b>	<b>18</b>
5.1	Gestione di progetto . . . . .	18
5.1.1	Pianificazione delle attività . . . . .	18
5.1.2	Coordinazione e controllo delle attività . . . . .	18
5.1.3	Gestione e controllo delle risorse . . . . .	18
5.1.4	Analisi e Gestione dei rischi . . . . .	18

5.1.5	Elaborazione dati . . . . .	19
5.1.6	Delega . . . . .	19
5.1.7	Responsabilità di sotto-progetto . . . . .	19
5.1.7.1	Assegnazione attività . . . . .	19
5.1.7.2	Gestione dei cambiamenti . . . . .	19
5.2	Analisi dei Requisiti . . . . .	19
5.2.1	Fattibilità . . . . .	19
5.2.2	Scoperta dei requisiti . . . . .	20
5.2.3	Interviste . . . . .	20
5.2.4	Riunioni interne e casi d'uso . . . . .	20
5.2.5	Classificazione e priorità . . . . .	20
5.2.6	Specifica . . . . .	21
5.2.7	Verifica dei requisiti . . . . .	22
5.3	Validazione <sub>g</sub> dei requisiti . . . . .	22
5.3.1	Interna . . . . .	22
5.3.2	Esterna . . . . .	22
5.4	Gestione delle modifiche ai requisiti . . . . .	22
5.5	Progettazione . . . . .	23
5.5.1	Specifica Tecnica . . . . .	23
5.5.1.1	Diagrammi UML . . . . .	23
5.5.1.2	Design pattern . . . . .	23
5.5.1.3	Tracciamento componenti . . . . .	23
5.5.1.4	Definizione di Prodotto . . . . .	23
5.5.1.5	Diagrammi UML . . . . .	23
5.5.1.6	Test di unità . . . . .	23
5.5.1.7	Test di integrazione . . . . .	24
5.6	Verifica . . . . .	24
5.6.1	Verifica dei documenti . . . . .	24
5.6.2	Verifica diagrammi UML . . . . .	25
5.6.3	Verifica del codice . . . . .	25
5.6.3.1	Analisi Statica . . . . .	25
5.6.3.2	Analisi Dinamica . . . . .	25
5.6.3.3	Validazione . . . . .	26
5.7	Codifica . . . . .	26
5.7.1	Nomi . . . . .	26
5.7.2	Documentazione . . . . .	26
<b>6</b>	<b>Ambiente di lavoro</b>	<b>27</b>
6.1	Risorse . . . . .	27
6.1.1	Risorse <sub>g</sub> necessarie: . . . . .	27
6.1.1.1	Risorse <sub>g</sub> umane . . . . .	27
6.1.1.2	Risorse <sub>g</sub> Hardware . . . . .	27
6.1.1.3	Risorse <sub>g</sub> software . . . . .	27
6.1.2	Risorse <sub>g</sub> disponibili . . . . .	27
6.1.2.1	Risorse <sub>g</sub> software . . . . .	28
6.2	Sistemi Operativi . . . . .	28

6.3	Coordinamento . . . . .	28
6.3.1	Software <sub>g</sub> di gestione del progetto . . . . .	28
6.3.2	Versionamento . . . . .	28
6.3.3	Software <sub>g</sub> di Integrazione Continua . . . . .	29
6.3.4	Condivisione dei file . . . . .	29
6.3.4.1	Google Drive . . . . .	29
6.3.5	Google Calendar . . . . .	30
6.4	Pianificazione . . . . .	30
6.5	Strumenti per i documenti . . . . .	30
6.5.1	LATEX . . . . .	30
6.5.2	Controllo ortografico . . . . .	30
6.5.3	Grafici UML . . . . .	30
6.5.4	Fogli di calcolo . . . . .	31
6.6	Strumenti per la codifica . . . . .	31
6.6.1	Stesura . . . . .	31
6.6.2	Verifica . . . . .	31
6.7	Protocollo per lo sviluppo dell'applicazione . . . . .	31
6.7.1	Creare un nuovo progetto . . . . .	31
6.7.2	Creazione ticket . . . . .	32
6.7.2.1	Ticket <sub>g</sub> di pianificazione . . . . .	32
6.7.2.2	Ticket <sub>g</sub> di realizzazione e controllo . . . . .	33
6.7.2.3	Ticket <sub>g</sub> di verifica . . . . .	34
6.7.2.4	Dipendenze temporali . . . . .	34
6.7.3	Aggiornamento ticket . . . . .	34
6.7.3.1	Ticket <sub>g</sub> di pianificazione . . . . .	35
6.7.3.2	Ticket <sub>g</sub> di realizzazione e controllo . . . . .	35
6.7.3.3	Ticket <sub>g</sub> di verifica . . . . .	36
6.7.4	Consigli di utilizzo . . . . .	38
6.8	LateTack . . . . .	39
6.8.1	Aggiunta nuovo requisito . . . . .	39

A	Lista di controllo	41
---	--------------------	----



## 1.1 Scopo del documento

- **Relazioni interpersonali** : comunicazione fra le varie figure professionali all'interno del gruppo di progetto;
- **Redazione documenti** : stili di redazione dei vari documenti interni e/o esterni;
- **Codifica**: stili e convenzioni di scrittura del codice sorgente;
- **Procedure di automazione**: strumenti e procedure per l'automazione di attività tecniche;
- **Definizione dell'ambiente di lavoro**: programmi utilizzati dall'intero gruppo di progetto.

## 1.2 Glossario

### 1.3 Riferimenti

- Piano di Progetto: [PianoDiProgetto\\_v.1.0.0.pdf](#);
- Piano di Qualifica: [PianoDiQualifica\\_v.1.0.0.pdf](#);



## 2 Ruoli di progetto

Durante lo sviluppo del progetto vi saranno diversi ruoli che i membri del gruppo andranno a ricoprire. Tali ruoli rappresentano figure aziendali specializzate, indispensabili per il buon esito del progetto. Ciascun componente del gruppo dovrà ricoprire almeno una volta ogni ruolo(vincolo organigramma). Si deve inoltre certificare che non vi siano conflitti di interesse nello svolgimento delle attività di verifica e di approvazione.

Per garantire che la rotazione dei ruoli non provochi conflitti è necessario che le attività di stesura e verifica vengano pianificate dettagliatamente e che i soggetti interessati rispettino i compiti a loro assegnati. Sarà poi compito del *Verificatore* controllare attentamente il diario delle modifiche di ogni documento per individuare eventuali incongruenze.

Si descrivono ora i diversi ruoli di progetto, con le relative responsabilità e le modalità operative affinché essi possano svolgere i compiti assegnati con l'ausilio dei software scelti per il progetto.

## 2.1 Responsabile di Progetto

Il *Responsabile di Progetto* rappresenta il progetto, in quanto accentra su di sé le responsabilità di scelta ed approvazione, ed il gruppo, in quanto presenta al committente i risultati del lavoro svolto. Detiene il potere decisionale, quindi la responsabilità su:

- Pianificazione, coordinamento e controllo delle attività;
- Gestione e controllo delle risorse;
- Analisi e gestione dei rischi;
- Approvazione dei documenti;
- Approvazione dell'offerta economica.

Di conseguenza, ha il compito di assicurarsi che le attività di verifica vengano svolte sistematicamente seguendo le *Norme di Progetto*, vengano rispettati i ruoli e le competenze assegnate nel *Piano di Progetto*, non vi siano conflitti di interesse tra redattori e verificatori. Egli è l'unico a poter decidere l'approvazione di un documento e a sancirne la distribuzione. Solo in casi particolari il *Responsabile* può delegare ad un verificatore l'approvazione di un documento come descritto nella sezione 5.1.6.

Ha inoltre l'incarico di gestire la creazione e l'assegnazione dei ticket delle macro-fasi e di assegnare ad un membro del gruppo il ruolo di responsabile di quest'ultima. Redige il *Piano di Progetto* e collabora alla stesura del *Piano di Qualifica*, in particolare nella sezione relativa alla pianificazione.

## 2.2 Amministratore

L'*Amministratore* è responsabile del controllo, dell'efficienza e dell'operatività dell'ambiente di lavoro. Le mansioni di primaria importanza che gli competono sono:

- Ricerca di strumenti che possano automatizzare qualsiasi compito che possa essere tolto all'umano;

- Risoluzione dei problemi legati alle difficoltà di gestione e controllo dei processi e delle risorse. La risoluzione di tali problemi richiede l'adozione di strumenti adatti;
- Controllo delle versioni e delle configurazioni del prodotto;
- Gestione dell'archiviazione e del versionamento della documentazione di progetto;
- Fornire procedure e strumenti per il monitoraggio e la segnalazione per il controllo qualità. Redige le *Norme di Progetto*, dove spiega e norma l'utilizzo degli strumenti, redige la sezione del *Piano di Qualifica* dove vengono descritti strumenti e metodi di verifica;

### 2.3 Analista

L'*Analista* è responsabile delle attività di analisi. Le responsabilità di spicco per tale ruolo sono:

- Produrre una specifica di progetto comprensibile, sia per il Proponente, sia per il Committente che per il *Progettista*, e motivata in ogni suo punto;
- Comprendere appieno la natura e la complessità del problema.

Redige lo *Studio di Fattibilità*, l'*Analisi dei Requisiti* e parte del *Piano di Qualifica*. Partecipa alla redazione del *Piano di Qualifica* in quanto conosce l'ambito del progetto ed ha chiari i livelli di qualità richiesta e le procedure da applicare per ottenerla.

## 2.4 Progettista

Il *Progettista* è responsabile delle attività di progettazione. Le responsabilità di tale ruolo sono:

- Produrre una soluzione attuabile, comprensibile e motivata;
- 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.

Redige la *Specifica Tecnica*, la *Definizione di Prodotto* e le sezioni inerenti le metriche di verifica della programmazione del *Piano di Qualifica*.

## 2.5 Verificatore

Il *Verificatore* è responsabile delle attività di verifica. Ha il compito di effettuare la verifica dei documenti utilizzando gli strumenti e i metodi proposti dal *Piano di Qualifica* e attenendosi a quanto descritto nelle *Norme di Progetto*. Le responsabilità di tale ruolo sono:

- Assicurare che l'attuazione delle attività sia conforme alle norme stabilite;
- Controllare la conformità di ogni stadio del ciclo di vita del prodotto.

Redige la sezione del *Piano di Qualifica* che illustra l'esito e la completezza delle verifiche e delle prove effettuate.

## 2.6 Programmatori

Il *Programmatore* è responsabile delle attività di codifica e delle componenti di ausilio necessarie per l'esecuzione delle prove di verifica e validazione. Le responsabilità di tale ruolo sono:

- Implementare rigorosamente le soluzioni descritte dal *Progettista*, da cui seguirà quindi la realizzazione del prodotto;
- Scrivere codice documentato, versionato, manutenibile e che rispetti gli standard stabiliti per la scrittura del codice;
- Implementare i test sul codice scritto, necessari per prove di verifica e validazione. Redige il *Manuale Utente* e produce una abbondante documentazione del codice.

### 3 Collaborazione

### 3.1 Comunicazioni

### 3.1.1 Comunicazioni interne

Per le comunicazioni interne è stato aperto un gruppo privato su Facebook accessibile ai singoli membri del team.

<https://www.facebook.com/groups/1709354699290988>

Inoltre ogni membro del team dovrà annotare i propri impegni sullo strumento Google Calendar, il quale verrà utilizzato per segnare qualsiasi tipo di impegno: di gruppo e individuale.

In caso di comunicazioni vocali o videoconferenze verrà utilizzato Skype.

### 3.1.2 Comunicazioni esterne

Per quanto riguarda le comunicazioni esterne (verso Committente<sub>g</sub> e/o Proponente<sub>g</sub>) è stata creata una casella di posta elettronica dedicata gestita dal Responsabile di progetto<sub>g</sub>:

latebutsafe@gmail.com

è compito del Responsabile gestire le informazioni in entrata e in uscita avvisando il proprio gruppo e il committente<sub>g</sub>/proponente<sub>g</sub> di eventuali comunicazioni rispettivamente in entrata e in uscita.

## 3.2 Riunioni

### 3.2.1 Interne

- Ogni membro del gruppo può richiedere una riunione interna tramite un post all'interno del gruppo di Facebook (tramite l'uso del tag  $_{\text{g}}$  [Richiesta Riunione Interna  $x$ ] con  $x$  numero incrementato di 1 rispetto alla richiesta precedente). Questa richiesta in base alle risposte degli altri componenti verrà presa in esame dal Responsabile;
- Una volta valutate le motivazioni della richiesta il Responsabile controlla sul calendario del gruppo le disponibilità dei vari componenti;
- Il Responsabile entro 1 giorno lavorativo pubblica una nuova discussione con tag  $_{\text{g}}$  [Esito Richiesta Interna  $x$ ], in cui, in caso positivo annuncia orario e luogo della riunione, in caso negativo annulla o rimanda la richiesta al successivo incontro;
- Nel caso in cui, per diversi motivi, alla riunione non potessero presenziare più di due membri, si procede a fissare una nuova riunione (vedi punto 2 e seguenti).

### 3.2.1.1 Casi Particolari

Per le richieste di riunioni interne vicine (cinque giorni lavorativi) ad una milestone<sub>g</sub>, se approvate dal Responsabile, verranno indette il giorno stesso o il seguente.

### 3.2.2 Esterne

Per le riunioni esterne (quindi gli incontri con il Proponente/Committente<sub>g</sub>) la prassi è la medesima delle riunioni interne; può essere avanzata da qualsiasi membro del gruppo con il tag<sub>g</sub> [Richiesta Riunione esterna *x*]. In questo caso il Responsabile avrà il duplice compito di valutare la richiesta dopo aver consultato il calendario e di contattare il committente<sub>g</sub>, per accordarsi su tempi e luogo dell'incontro, che verranno poi riferiti sulla piattaforma di comunicazioni interne tramite il tag<sub>g</sub> [Esito Richiesta Riunione Esterna *x*].

### 3.2.3 Esito

Ad ogni riunione (sia interna che esterna) il Responsabile ha il dovere di assicurarsi che venga redatto un verbale che riassume gli argomenti trattati durante l'incontro e tutte le eventuali decisioni prese; i membri del gruppo hanno l'obbligo di applicare le eventuali modifiche o correzioni decise durante la riunione ed è del responsabile il dovere che i problemi emersi durante il verbale siano stati risolti.

### 3.3 Repository e strumenti per la condivisione di file

### 3.3.1 Repository

Sono stati creati due repository Git:

- documents.git : disponibile all'indirizzo

<https://github.com/PetrucciMauro/documents>

conterrà i sorgenti L<sup>A</sup>T<sub>E</sub>X e gli script necessari alla stesura dei documenti;

- source.git : disponibile all'indirizzo

<https://github.com/PetrucchiMauro/source>

conterrà i sorgenti dell'applicazione.

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. Il meccanismo di verifica e approvazione è descritto in dettaglio nella sezione 5.6.1.

### 3.3.2 Condivisione file

Per la condivisione informale di file e per il lavoro collaborativo su documenti di supporto, si usa la piattaforma di condivisione file online Google Drive. Trattandosi di strumenti informali, non si definiscono procedure rigorose d'uso e se ne lascia la descrizione alle sezioni 6.3.4.



## 4 Documentazione

Questo capitolo descrive tutte le convenzioni scelte ed adottate dai LateButSafe riguardo alla stesura, verifica e approvazione della documentazione da produrre.

## 4.1 Template

Tutti i documenti devono essere realizzati utilizzando un template<sub>g</sub> L<sup>A</sup>T<sub>E</sub>X. Onde evitare modifiche manuali che farebbero perdere molto tempo, Il nome dei file<sub>g</sub> deve rispondere alla seguente formattazione senza spazi: “[nome documento]-[versione]”. La parte della versione deve riportare la dicitura “v.” seguita dal numero di versione (ad es: NormeDiProgetto\_v.1.0.0.pdf). Tale modello si può trovare nel repository in documents/template.

## 4.2 Contenuto e struttura dei documenti

Ogni documento ufficiale deve essere composto dalle seguenti sezioni:

- Prima pagina: deve riportare titolo, logo ed informazioni del documento;
- Breve prefazione;
- Registro delle modifiche;
- Indice del documento;
- Indice di figure e tabelle (se presenti);
- Introduzione;
- Corpo.

Ogni pagina deve avere un'intestazione e un piè di pagina:

- **Intestazione:** logo del gruppo e nome del documento;
- **Piè di pagina:** versione documento, università e anno accademico, numeri di pagina e licenza.

### 4.2.1 Verballi

Per quanto riguarda i verbali degli incontri, essi devono essere redatti dal Responsabile di Progetto<sub>g</sub> ad ogni riunione. Esso deve rispettare la formattazione regolata alla sezione 4.3 e successive ma è da considerarsi solo come promemoria per il gruppo.

Il nome di ogni Verbale deve rispettare la seguente dicitura: “Verbale\_[tipo incontro]-[data]”  
dove il tipo incontro può essere di due tipi:

- Interno (INT): incontro effettuato tra i membri del gruppo;
- Esterno (EXT): incontro effettuato tra i membri del gruppo e committente<sub>g</sub> e/o proponente<sub>g</sub>.

La prima pagina di ogni verbale deve obbligatoriamente contenere i seguenti campi, in ordine:

- Data;
- Luogo secondo il formato “[città],[provincia],[sede]”;
- Ora<sub>g</sub> secondo il formato “dalle ore [hh]:[mm] alle ore [hh]:[mm]” dove hh indica le ore e mm i minuti i quali vanno espressi nel formato 24 ore secondo lo standard ISO<sub>g</sub> 8601:2004;
- Partecipanti interni al gruppo elencandoli rispettando il formato “[nome] [cognome][, [...]]”;
- Partecipanti esterni al gruppo rispettando il formato “[nome] [cognome][ruolo][, [...]]” in cui il ruolo può essere Committente<sub>g</sub> oppure Proponente<sub>g</sub>;
- Contenuto dell’incontro;
- Firme: devono essere comprese quelle di tutti i partecipanti del gruppo LateButSafea conferma della presa visione del documento.

### 4.2.2 Lettera di presentazione

La lettera di presentazione deve contenere

- Logo del gruppo;
- Intestazione nel seguente formato:  
Prof. Tullio Vardanega  
Università degli Studi di Padova  
Via Trieste 63  
35121 Padova (PD)
- Breve introduzione (facoltativa);
- Elenco di tutti i documenti in consegna;
- Varie ed eventuali, osservazioni (facoltative);
- Firma del responsabile nel seguente formato:  
Nome Cognome  
il Responsabile del gruppo LateButSafe  
Firma del responsabile

### 4.3 Norme tipografiche

Per rendere la documentazione organizzata, leggibile e standard abbiamo adottato le forme testuali riportate di seguito.

- **Carattere:** il carattere dovrà avere come dimensione minima 12. Per l’inserimento di linee di codice<sub>g</sub> il carattere da utilizzare dovrà essere di tipo Monospace;
- **Grassetto:** da utilizzare maggiormente per definire i titoli e dare una panoramica generale del testo ed in maniera minore per sottolineare passaggi importanti e parole chiave;



- **Corsivo:** da utilizzare per riportare citazioni da fonti esterne o riferimenti;
- **Sottolineato:** da utilizzare all'interno del testo per dare importanza a determinati concetti;
- **Maiuscolo:** deve essere limitato all'indicazione di acronimi e nei casi specificati nei Formati di Riferimento (4.4);
- **Punteggiatura:** adottare la formattazione standard ossia la punteggiatura deve precedere sempre un carattere di spazio e non viceversa;
- **Lettera maiuscola:** deve seguire esclusivamente un punto, un punto esclamativo o un punto interrogativo;
- **Parentesi:** una qualsiasi frase racchiusa fra parentesi non deve iniziare con un carattere di spaziatura e non deve chiudersi con un carattere di punteggiatura e/o di spaziatura;
- **Elenchi puntati o numerati:** ogni elemento 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<sub>g</sub>);
- **Glossario:** le parole accompagnate da (g) in pedice sono quelle che presentano una corrispondenza nel Glossario;
- **Pagine:** è obbligatorio porre i numeri di pagina in ogni documento nel formato n di totale pagine e mantenere i margini fissati dal template<sub>g</sub> di cui sopra (4.2).

## 4.4 Formati di riferimento e altro

Per quanto riguarda i riferimenti, è opportuno rispettare le seguenti indicazioni:

- percorsi<sub>g</sub>: per gli indirizzi<sub>g</sub> web<sub>g</sub> completi e indirizzi<sub>g</sub> e-mail deve essere utilizzato il comando appositamente fornito da L<sup>A</sup>T<sub>E</sub>X:  
`\urlg {Percorso};`
- Ancore: i riferimenti alle sezioni interne del medesimo documento devono essere scritte utilizzando il comando fornito da L<sup>A</sup>T<sub>E</sub>X: `\ref {label da riferire}`.

La **Data** deve essere espressa, seguendo lo standard ISO<sub>g</sub> 8601:2004, nel formato: AAAA-MM-GG (AAAA rappresenta l'anno in quattro cifre, MM il mese in due cifre e GG il giorno in due cifre).

Le **Abbreviazioni** ammesse sono le seguenti e valgono per tutti i documenti:

- **AR:** Analisi dei Requisiti<sub>g</sub>;
- **GL:** Glossario;
- **NP:** Norme di Progetto<sub>g</sub>;
- **PQ:** Piano di Qualifica;



- **PP**: Piano di Progetto<sub>g</sub>;
- **SF**: Studio di Fattibilità;
- **RR**: Revisione dei Requisiti<sub>g</sub>;
- **RP**: Revisione di Progettazione;
- **RQ**: Revisione di Qualifica;
- **RA**: Revisione di Accettazione.

I **Nomi ricorrenti** nei vari documenti devono rispettare le seguenti indicazioni:

- Ruoli di progetto<sub>g</sub> e nomi dei documenti: devono essere formattati utilizzando la prima lettera maiuscola di ogni parola che non sia una preposizione (es. Responsabile di Progetto<sub>g</sub>);
- Nomi dei file<sub>g</sub>: il riferimento deve essere comprensivo dell'estensione<sub>g</sub> del file<sub>g</sub> e formattato in corsivo;
- Nomi propri: l'utilizzo dei nomi propri deve seguire il formalismo Cognome Nome;
- Nome del gruppo: deve essere sempre espresso nel formato: LateButSafe;
- Nome del progetto<sub>g</sub>: deve essere sempre espresso nel formato: Premi.

## 4.5 Immagini e tabelle

Tutte le immagini devono essere in formato JPG, PNG o PDF mentre ogni tabella deve rispettare il formato  $\text{\LaTeX}$ .

Ogni figura o tabella inserita deve avere una breve didascalia composta da un identificativo numerico univoco seguito, ove sia ritenuto necessario, da una breve descrizione. La numerazione di immagini e tabelle sarà attribuita da L<sup>A</sup>T<sub>E</sub>X.

## 4.6 Glossario

Il glossario è unico per tutti i documenti e deve essere organizzato come definito nella sezione Documenti 4.1. Tutti i membri del gruppo possono modificarlo.

I termini all'interno del glossario saranno organizzati nel seguente modo:

- Tutti termini saranno in ordine alfanumerico;
- Tutti i termini devono essere in grassetto e iniziare con la lettera maiuscola , la definizione del termine sarà preceduta dal carattere ”:” ;
- Tutti i termini devono fornire chiarimenti su concetti che possono essere confusi quindi non devono essere inseriti termini il cui significato è già noto.



Il file<sub>g</sub> relativo al Glossario è il seguente: [Glossario\\_v.1.0.0.pdf](#)



Dopo aver descritto i ruoli di progetto e le relative funzioni, si procede ora ad elencare le procedure che essi devono seguire in modo rigoroso per convergere agli obiettivi posti nel *Piano di Qualifica*.

Le responsabilità di gestione dell'intero progetto, dalla nascita alla conclusione, sono da attribuire al *Responsabile di Progetto*. Quest'ultimo dovrà garantire un corretto sviluppo delle attività utilizzando, qualora sia possibile, degli strumenti che gli consentano di:

- ### 5.1.1 Pianificazione delle attività

Per pianificare le attività il *Responsabile di Progetto* deve realizzare un diagramma di Gantt per ciascuna fase indicata nella sezione Pianificazione del [PianoDiProgetto\\_v.1.0.0.pdf](#), utilizzando OmniPlan come descritto nella sezione 6.4.

Per coordinare e controllare le attività il Responsabile di Progetto deve riportare la struttura creata con OmniPlan in Redmine sfruttando il suo sistema di ticketing come descritto nella sezione 6.7. In questo modo ciascun componente del gruppo sarà avvisato delle attività ad esso assegnate e potrà inserire lo stato delle stesse permettendo al Responsabile di verificare immediatamente l'avanzamento del progetto.

Per gestire e controllare le risorse il Responsabile di Progetto deve utilizzare Redmine come indicato nella sezione 6.7 che gli consente anche di verificare l'avanzamento di ogni processo come riportato nel [PianoDiProgetto v.1.0.0.pdf](#).

Durante l'avanzamento del progetto il *Responsabile di Progetto* deve monitorare costantemente il verificarsi dei rischi descritti nel [PianoDiProgetto\\_v.1.0.0.pdf](#) ed eventuali nuovi rischi, attuando le contromisure descritte e riportando gli effettivi riscontri.

### 5.1.5 Elaborazione dati

Il Responsabile di Progetto deve sfruttare i fogli di calcolo elettronico, come descritto nella sezione 6.5.4, per elaborare i dati raccolti durante lo sviluppo del progetto e riportarli nel *Piano di Progetto*.

### 5.1.6 Delega

Il *Responsabile di Progetto*, nel caso in cui abbia redatto una parte di un documento, può delegare l'approvazione di tale documento ad un Verificatore.

### 5.1.7 Responsabilità di sotto-progetto

Ogni macroattività può essere assegnata dal Responsabile ad un responsabile di sottoprogetto, i cui compiti saranno l'assegnazione delle singole attività alle risorse rese disponibili e la gestione dei cambiamenti.

#### 5.1.7.1 Assegnazione attività

Per assegnare attività alle risorse disponibili, il responsabile di sotto-progetto dovrà seguire le procedure di ticketing descritte in 6.7.2.2.

#### 5.1.7.2 Gestione dei cambiamenti

In caso di errori, in seguito alla notifica da parte del *Verificatore* tramite ticket, il responsabile di sotto-progetto dovrà assegnare la correzione mediante la procedura di ticketing descritta in 6.7.2.2. Al termine della correzione, sarà compito del responsabile di sotto-progetto accettare o respingere la modifica, e richiederne di conseguenza il rifacimento. Nel caso la correzione riguardi un'attività di codifica, sarà compito del responsabile di sotto-progetto programmare una nuova esecuzione dei test di unità e di integrazione correlati al modulo modificato.

## 5.2 Analisi dei Requisiti

### 5.2.1 Fattibilità

A partire da informazioni preliminari sul capitolato, lo studio di fattibilità dovrà generare un rapporto che indichi la convenienza o meno del gruppo nello sviluppo del sistema. In particolare si dovrà considerare:

1. Sufficienza di risorse umane<sub>g</sub>;
2. Rapporto tra i costi ed i benefici;
3. Rischi individuati.

Nello stimare i benefici dovrà essere data molta importanza alle competenze che i membri del gruppo acquisirebbero nello sviluppo del sistema.

### 5.2.2 Scoperta dei requisiti

### 5.2.3 Interviste

al fine di evitare interviste infruttuose verrà preparato un elenco di punti da sottoporre al proponente<sub>g</sub> in modo da dare una direzione precisa all'intervista. Potrebbe essere utile discutere con il proponente<sub>g</sub> dei casi d'uso<sub>g</sub> analizzati internamente al gruppo durante la fase di analisi. Le richieste di interviste al proponente<sub>g</sub> avverranno con le modalità descritte in "comunicazioni esterne". Durante ogni intervista dovrà essere scritta una minuta che sarà confermata dal proponente<sub>g</sub>, eventualmente con le opportune modifiche. La minuta sarà confermata al termine dell'incontro. Quando non fosse un problema per il proponente<sub>g</sub> l'audio dell'intervista dovrà essere registrato per favorire la futura fase di analisi.

#### 5.2.4 Riunioni interne e casi d'uso

Individuale e durante le riunioni interne gli analisti dovranno analizzare le informazioni raccolte dalle interviste con il proponente<sub>g</sub> per individuare problemi e fonti da cui attingere i requisiti<sub>g</sub>.

L'individuazione dei requisiti<sub>g</sub> funzionali sarà guidata dai casi d'uso. I casi d'uso potranno avere rappresentazione a diagrammi ma ogni caso d'uso dovrà avere anche la rappresentazione testuale. In particolare nella rappresentazione testuale si definirà:

1. Identificativo;
2. Attore primario;
3. Precondizioni;
4. Postcondizioni;
5. Scenario principale;
6. estensioni<sub>g</sub>.

Per la sintassi si rimanda a "Dall'idea al codice<sub>g</sub> con UML2.0, Luciano Baresi, Luigi Lavazza, Massimiliano Pianciamore".

### 5.2.5 Classificazione e priorità

I requisiti dovranno essere classificati in:

1. Requisiti<sub>g</sub> di processo<sub>g</sub>;
2. Requisiti<sub>g</sub> di prodotto.

I requisiti di prodotto saranno classificati in base a:

1. Tipologia;
2. Importanza;
3. Provenienza.

Dove i gradi di importanza saranno:

- **{Obbligatorio}**: requisito<sub>g</sub> obbligatorio;
- **{Desiderabile}**: requisito<sub>g</sub> desiderabile;
- **{Opzionale}**: requisito<sub>g</sub> opzionale.

La provenienza può essere:

- **{Capitolato}**: da capitolato;
- **{Interni}**: da analisi interna;
- **{Proponente}**: da incontro con proponente.

Mentre le tipologie saranno:

- $\{\mathbf{RF}\}$ : requisito<sub>g</sub> funzionale;
- $\{\mathbf{RQ}\}$ : requisito<sub>g</sub> di qualità;
- $\{\mathbf{RV}\}$ : requisito<sub>g</sub> di vincolo.

### 5.2.6 Specifica

Nella specifica dei requisiti<sub>g</sub> dovrà essere considerato come riferimento lo standard IEEE 830-1998. In particolare saranno da perseguire le seguenti caratteristiche dei requisiti<sub>g</sub>:

1. Non ambigui;
2. Corretti;
3. Completi;
4. Verificabili;
5. Consistenti;
6. Modificabili;
7. Tracciabili;
8. Ordinati per rilevanza.

I requisiti<sub>g</sub> dovranno essere specificati in un documento "Analisi dei requisiti" secondo la struttura definita nello standard IEEE 830-1998. La specifica dei requisiti<sub>g</sub> dovrà essere documentata in forma tabellare per evitare ambiguità. Per ogni requisito<sub>g</sub> dovranno essere definiti un codice<sub>g</sub>, una descrizione, un riferimento alla fonte e un riferimento alla verifica. Al fine di rendere meno ambigui i requisiti<sub>g</sub> sarà redatto un "Glossario" contenente la definizione di tutti i termini non ovvi usati in fase di analisi.

### 5.2.7 Verifica dei requisiti

Per ogni requisito<sub>g</sub> di processo<sub>g</sub> specificato dovrà essere presente in "Piano di qualifica" un riferimento alle sezioni di "Norme di progetto" in cui viene assicurato il soddisfacimento del requisito<sub>g</sub>. Per ogni requisito<sub>g</sub> di prodotto specificato dovrà essere descritto brevemente il metodo che verrà usato per verificarne il soddisfacimento.

Per favorire la tracciabilità tra requisiti<sub>g</sub> e metodi di verifica dovrà essere presente in "Piano di qualifica" una tabella in cui si definiscono: codice<sub>g</sub> di requisito<sub>g</sub>, codice<sub>g</sub> di verifica e modalità di verifica. Se il requisito<sub>g</sub> è di processo<sub>g</sub>, la modalità di verifica conterrà i riferimenti alle sezioni corrispondenti in "Norme di progetto".

### 5.3 Validazione<sub>g</sub> dei requisiti

### 5.3.1 Interna

Saranno verificate la correttezza e la completezza dei requisiti<sub>g</sub> rispetto ai bisogni. Ciò verrà fatto tramite tracciamento tra specifica dei requisiti<sub>σ</sub> e bisogni individuati.

Saranno verificate la correttezza e la completezza dei metodi di verifica dei requisiti<sub>g</sub> rispetto ai requisiti<sub>g</sub>. Ciò verrà fatto tramite tracciamento tra specifica dei requisiti<sub>g</sub> e metodi di verifica.

### 5.3.2 Esterna

Terminata la validazione<sub>g</sub> interna verranno presentati al proponente<sub>g</sub> i documenti "Analisi dei requisiti" e "Piano di qualifica", se accettati costituiranno una baseline per la fase successiva del progetto<sub>g</sub> altrimenti verranno gestite le richieste di modifica secondo i metodi descritti in "Gestione dei cambiamenti".

## 5.4 Gestione delle modifiche ai requisiti

A tutte le proposte di modifica dei requisiti, dovrà essere applicata la seguente procedura:

1. Deduzione, analisi e specifica dei cambiamenti;
2. Stima dei costi del cambiamento considerando quante modifiche dovranno essere fatte ai requisiti<sub>g</sub> e al progetto<sub>g</sub> del sistema;
3. Decisione ed eventuale implementazione del cambiamento nei requisiti<sub>g</sub> e nel progetto<sub>g</sub> di sistema.

Per gestire i cambiamenti e per facilitare il tracciamento dei requisiti<sub>g</sub> verrà usato un software<sub>g</sub> appositamente creato dal gruppo. L'amministratore avrà il compito di gestire il server<sub>g</sub> e amministrare i diritti di accesso degli utenti alle funzionalità fornite. In particolare gli analisti dovranno usare i modelli definiti all'inizio della fase di analisi. Per evitare problemi dovuti a modifiche concorrenti alla base dati l'amministratore dovrà garantire che ad ogni istante solo un analista possa modificare un certo sotto albero della foresta dei requisiti<sub>r</sub> e dei test.

## 5.5 Progettazione

### 5.5.1 Specifica Tecnica

I Progettisti devono descrivere la progettazione ad alto livello dell'architettura dell'applicazione e dei singoli componenti nella *Specifica Tecnica* e provvedere alla progettazione di opportuni test di integrazione.

### 5.5.1.1 Diagrammi UML

Devono essere realizzati i seguenti diagrammi:

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

### 5.5.1.2 Design pattern

I *Progettisti* devono descrivere i design pattern utilizzati per realizzare l'architettura: di essi si deve includere una breve descrizione e un diagramma che ne esemplifichi il funzionamento e la struttura.

### 5.5.1.3 Tracciamento componenti

Ogni requisito deve essere tracciato al componente che lo soddisfa. Il software LateTrack genera automaticamente le tabelle di tracciamento come descritto nella sezione 6.8. In questo modo sarà possibile misurare il progresso nell'attività di progettazione e garantire che ogni requisito venga soddisfatto.

#### 5.5.1.4 Definizione di Prodotto

I Progettisti devono produrre la Definizione di Prodotto dove viene descritta la progettazione di dettaglio del sistema ampliando quanto scritto nella Specifica Tecnica.

### 5.5.1.5 Diagrammi UML

Devono essere aggiornati i seguenti diagrammi:

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

#### 5.5.1.6 Test di unità

I Progettisti dovranno definire i test d'unità necessari per verificare che le componenti del sistema funzionino nel modo previsto.





I Progettisti devono definire delle classi di verifica necessarie per verificare che le componenti del sistema funzionino nella maniera prevista.

L'obiettivo delle attività di verifica è quello di trovare e rimuovere i problemi presenti. Un problema può verificarsi a vari livelli, e per ogni livello assume un nome diverso:

- Fault (difetto): è l'origine del problema, ciò che fa scaturire il malfunzionamento;
- Error (errore): è lo stato per cui il software<sub>g</sub> si trova in un punto sbagliato del flusso di esecuzione o con valori sbagliati rispetto a quanto previsto dalla specifica;
- Failure (fallimento, guasto): è un comportamento difforme dalla specifica, cioè la manifestazione dell'errore all'utente del software<sub>g</sub>.

Esiste una relazione di causa-effetto fra questi tre termini:

*DIFETTO*  $\rightarrow$  *ERRORE*  $\rightarrow$  *FALLIMENTO*

Non sempre un errore dà origine ad un fallimento: ad esempio potrebbero esserci alcune variabili che si trovano in stato erraneo ma non vengono lette, o non viene percorso<sub>g</sub> il ramo di codice<sub>g</sub> che le contiene.

E' necessario prestare particolare attenzione a questo tipo di errori (detti anche quiescenti), avvalendosi anche di strumenti per il rilevamento dei bug.

Ogni qualvolta avvenga un cambiamento sostanziale nello sviluppo del prodotto, si istanzierà il processo<sub>v</sub> di verifica.

Nello specifico durante ogni fase (Analisi, Progettazione, Realizzazione e Validazione<sub>g</sub>) saranno applicate le tecniche di verifica qui descritte nei seguenti casi:

- Conclusione della prima redazione di un documento;
- Conclusione della prima redazione di un file<sub>g</sub> di codice<sub>g</sub>;
- Conclusione della modifica sostanziale di un documento: quando il versionamento passa da .x.y.z a .x.y+1.0 oppure a .x+1.0.0.

Per eseguire un'accurata verifica dei documenti redatti è necessario seguire il seguente protocollo:

1. **Controllo sintattico e del periodo:** Utilizzando TeXstudio e GNU Aspell vengono evidenziati e corretti gli errori di grammatica più evidenti. Gli errori di sintassi, di sostituzione di lettere che provocano la creazione di parole grammaticalmente corrette ma sbagliate nel contesto ed i periodi di difficile comprensione necessitano dell'intervento di un verificatore umano. Per questa ragione ciascun documento dovrà essere sottoposto ad un walkthrough da parte dei verificatori per individuare tali errori;



- ### 5.6.2 Verifica diagrammi UML

### 5.6.3 Verifica del codice

#### 5.6.3.1 Analisa Statika

- #### 5.6.3.2 Analisis Dinamica

Università degli studi di Padova - 2014/2015

#### 5.6.3.3 Validazione

La validazione<sub>g</sub> del codice<sub>g</sub> HTML e CSS<sub>g</sub> dell'applicazione da noi sviluppata verrà fatta tramite il servizio W3C<sub>g</sub> Validator32 del W3C<sub>g</sub>.

## 5.7 Codifica

Le convenzioni di codifica che tutti i membri del gruppo devono seguire sono quelle specificate alla seguente pagina :

[http://www.w3schools.com/js/js\\_conventions.asp](http://www.w3schools.com/js/js_conventions.asp)

### 5.7.1 Nomi

I nomi di variabili, classi, funzioni, metodi e commenti dovranno essere in camelCase. I nomi di variabili, metodi e funzioni dovranno avere la prima lettera minuscola. I nomi di variabili globali e costanti dovranno essere in UPPERCASE.

### 5.7.2 Documentazione

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

```
1  /*!  
2  * \file Nome del file  
3  * \author Autore (indirizzo email dell'autore)  
4  * \date Data di creazione  
5  * \brief Breve descrizione del file  
6  *  
7  * Descrizione dettagliata del file  
8  */  
9  
10 Prima di ogni metodo dovrà essere presente un commento  
11     contenente:  
12  
13 /*!  
14 * \brief Breve descrizione della funzione  
15 * \param Nome del primo parametro  
16 * \param Nome del secondo parametro  
17 * \return Valore ritornato dalla funzione  
18 */
```



## 6 Ambiente di lavoro

### 6.1 Risorse

#### 6.1.1 Risorse<sub>g</sub> necessarie:

##### 6.1.1.1 Risorse<sub>g</sub> umane

I ruoli necessari a garantire la qualità del prodotto sono:

- Responsabile di Progetto<sub>g</sub>;
- Amministratore;
- Verificatore;
- Programmatore.

##### 6.1.1.2 Risorse<sub>g</sub> Hardware

Saranno necessari:

- Computer con installato software<sub>g</sub> necessario allo sviluppo del progetto<sub>g</sub> in tutte le sue fasi;
- Luoghi in cui svolgere riunioni, preferibilmente dotato di connessione ad Internet.

##### 6.1.1.3 Risorse<sub>g</sub> software

Saranno necessari:

- Strumenti per automatizzare i test;
- Framework<sub>g</sub> per eseguire test di unità;
- Piattaforma di versionamento per la creazione e gestione di ticket<sub>g</sub>;
- Debugger per i linguaggi di programmazione scelti;
- Browser<sub>g</sub> come piattaforma di testing dell'applicazione da sviluppare;
- Strumenti per effettuare l'analisi statica<sub>g</sub> del codice<sub>g</sub> per misurare le metriche<sub>g</sub>.

### 6.1.2 Risorse<sub>g</sub> disponibili

Sono disponibili:

- Computer personali dei membri del gruppo;
- Computer presenti nelle aule informatiche del Dipartimento di Matematica;
- Aule disponibili per incontri nel Dipartimento di Matematica;
- Un dispositivo Raspberry Pi 2 Model B, utilizzato come server<sub>g</sub> per programmi<sub>g</sub> organizzativi e di testing.



#### 6.1.2.1 Risorse<sub>g</sub> software

- Strumenti per il coordinamento 6.3;
- Strumenti per i documenti 6.5;
- Strumenti per la codifica 6.6;
- Strumenti verifica 5.6.2.

## 6.2 Sistemi Operativi

L'intero sviluppo del progetto<sub>g</sub> viene svolto in ambienti Unix-Like e Windows<sub>g</sub>, nello specifico, Ubuntu<sub>g</sub>, Mac, Windows<sub>g</sub>. Tale scelta è maturata dopo aver appurato che le tecnologie utilizzate per lo sviluppo del progetto<sub>g</sub> sono indipendenti dall'ambiente di sviluppo e di impiego.

## 6.3 Coordinamento

è stato predisposto un server<sub>g</sub> dedicato sul quale sono installate alcune applicazioni web<sub>g</sub> che facilitano la gestione del progetto<sub>g</sub>. Per connettersi al server<sub>g</sub>, l'indirizzo<sub>g</sub> è il seguente:

<http://gioberry.no-ip.org/>

### 6.3.1 Software<sub>g</sub> di gestione del progetto

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

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

### 6.3.2 Versionamento

Come strumento di versionamento si è deciso di utilizzare **Git**. Git è uno strumento di versionamento veloce e di facile apprendimento che rappresenta uno dei migliori strumenti attualmente esistenti.

Per lo sviluppo collaborativo abbiamo deciso di appoggiarci al servizio **Github** che fornisce non solo un repository<sub>g</sub> Git, ma anche strumenti utili alla collaborazione fra più persone, come il servizio di **Ticket**, **Wiki** e **Milestone**.

Per quanto riguarda l'uso di Git sui computer di sviluppo, si è deciso l'uso della versione ufficiale rilasciata dal team di sviluppo di Git(2.3.3).

Per interfacciarsi con il repository<sub>g</sub> viene utilizzato **SmartGit**, un client multi piattaforma che permette di utilizzare Git in maniera rapida.

Si descrive ora la procedura di corretto utilizzo del programma, SmartGit.

- **Clonare il repository**: è possibile clonare il repository<sub>g</sub> remoto in locale attraverso la seguente procedura:
  - Premere nel menu in alto il pulsante Repository<sub>g</sub> e successivamente Clone;
  - Nel riquadro comparso, inserire il link<sub>g</sub> del repository  
<https://github.com/PetrucciMauro/documents.git>  
oppure  
<https://github.com/PetrucciMauro/source.git>  
successivamente premere il pulsante next;
  - Tenere la schermata successiva con entrambi i box spuntati e premere next;
  - Selezionare la posizione in cui verrà salvata la versione locale del repository<sub>g</sub>.
- **Sincronizzare il repository** : Dalla schermata principale premere il pulsante pull;
- **Salvare una modifica in locale**: Dalla schermata principale premendo il pulsante commit<sub>g</sub> e inserendo nell'apposita textbox un Messaggio di commit, si salvano le modifiche effettuate ai file<sub>g</sub>;
- **Inviare le modifiche al repository<sub>g</sub> remoto**: Dalla schermata principale premere il pulsante Push e, successivamente alla comparsa del nuovo riquadro, ancora push, ciò comporterà l'invio delle modifiche ai file<sub>g</sub> al repository<sub>g</sub> remoto.

### 6.3.3 Software<sub>g</sub> di Integrazione Continua

Si è scelto di adottare **Travis** per applicare l'integrazione continua allo sviluppo del progetto<sub>g</sub>. Tale software<sub>g</sub> permette di pianificare ed eseguire dei compiti da eseguire sui file<sub>g</sub> sorgente. Mette inoltre a disposizione un cruscotto su cui è possibile visualizzare lo stato del codice<sub>g</sub> prodotto. Tale software<sub>g</sub> è infatti in grado di interagire con il software<sub>g</sub> di versionamento, e se disponibile con software<sub>g</sub> per l'esecuzione di test sul codice<sub>g</sub> prodotto.

### 6.3.4 Condivisione dei file

Si è inoltre scelto di utilizzare degli strumenti online che permettono di condividere file<sub>g</sub> in modo semplice e veloce e che consentono di organizzare gli appuntamenti personali dei singoli componenti del gruppo.

#### 6.3.4.1 Google Drive

In questa piattaforma di condivisione file<sub>g</sub> verranno salvati i documenti che:

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

Questo strumento dovrebbe permettere a 2 o più componenti del gruppo di interagire lavorando sugli stessi documenti contemporaneamente. Google Drive viene utilizzato come strumento di supporto allo sviluppo della documentazione e del software, presente su Git .

### 6.3.5 Google Calendar

Google Calendar viene utilizzato all'interno del gruppo per gestire le risorse umane<sub>g</sub>. In particolare 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.

## 6.4 Pianificazione

Per pianificare le attività legate allo sviluppo del progetto e la gestione delle risorse si è scelto di utilizzare OmniPlan. OmniPlan è un programma per il project management. Tale software è stato scelto in quanto possiede le seguenti caratteristiche:

- Facilità d'uso;
- Generazione dei diagrammi di gant;
- Possibilità di esportare il progetto in più estensioni diverse.

## 6.5 Strumenti per i documenti

### 6.5.1 LATEX

Per la stesura dei documenti è stato scelto di utilizzare il sistema L<sup>A</sup>T<sub>E</sub>X.

Il motivo principale dietro a questa scelta è la facilità di separazione tra contenuto e formattazione: con `LATEX` è possibile definire l'aspetto delle pagine in un `fileg templateg` 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<sub>g</sub> esistenti consente di implementare funzionalità comuni in maniera semplice. L'estensibilità<sub>g</sub> di L<sup>A</sup>T<sub>E</sub>X può essere sfruttata per creare funzioni<sub>g</sub> 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<sub>g</sub>. Per la scrittura di documenti L<sup>A</sup>T<sub>E</sub>X l'editor<sub>g</sub> consigliato è **TeXstudio**.

### 6.5.2 Controllo ortografico

Il software<sub>g</sub> per il controllo ortografico è **Aspell** . Il programma<sub>g</sub> viene richiamato da linea di comando.

### 6.5.3 Grafici UML

Per la stesura dei grafici UML viene utilizzato il programma<sub>g</sub> **Visual Paradigm**. Il programma<sub>g</sub> viene utilizzato in licenza Community Edition la quale ne permette l'uso gratuito per fini non commerciali.

#### 6.5.4 Fogli di calcolo

L'utilizzo di fogli di calcolo elettronici quali Calc, Excel e Numbers è a discrezione del singolo componente in base alla propria piattaforma utilizzata. I fogli di calcolo vengono usati per:

- Grafici a torta per l'utilizzo delle risorse;
- Grafici a torta per il costo dedicato a ciascuna risorsa;
- Istogrammi per le ore assegnate ad ogni componente del gruppo;
- Tabelle per il confronto tra preventivo e consuntivo;
- Istogrammi per il confronto tra ore preventivate e ore realmente impiegate da ciascuna risorsa.

## 6.6 Strumenti per la codifica

### 6.6.1 Stesura

Per al stesura del codice HTML e javascript verrà usata l' IDE *Aptana*, la scelta sull'utilizzo di questa IDE si è basata sulle necessità del gruppo di un ambiente di sviluppo per applicazioni-web completo,gratuito e multiplatforma. *Aptana* presenta molti strumenti già integrati, come code assistant e debugger.

### 6.6.2 Verifica

L'analisi statica e l'analisi dinamica integrate in Travis vengono lanciate automaticamente ad ogni push di codice nel repository. Al verificatore è richiesto quindi soltanto di analizzare l'output del tool, visualizzabile dalla dashboard di Travis.

## 6.7 Protocollo per lo sviluppo dell'applicazione

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

La scelta di tale  $\text{software}_g$  è descritta nella sezione 6.3.1.

### 6.7.1 Creare un nuovo progetto

La creazione di un progetto<sub>g</sub> è compito del *Responsabile di Progetto*.

Un nuovo progetto<sub>g</sub> rappresenta una macro-attività caratterizzata da molte sotto-attività supervisionate da un responsabile.

Per creare un nuovo progetto<sub>g</sub>:

- Aprire **Progetti**;
- Selezionare **Nuovo progetto**;
- Assegnare un **Nome** breve ma significativo;



- Nel caso in cui si voglia creare un sotto-progetto<sub>g</sub> indicare il nome del progetto<sub>g</sub> padre dall'omonimo campo;
- **Identificativo:** scrivere in minuscolo ed indicare codice<sub>g</sub> della fase a cui si riferisce ;
- Lasciare inalterati gli altri campi.

### 6.7.2 Creazione ticket

I ticket<sub>g</sub> vengono creati da:

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

I ticket<sub>g</sub> possono essere di tre tipologie:

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

#### 6.7.2.1 Ticket<sub>g</sub> di pianificazione

- Selezionare **Nuova segnalazione** da menù principale;
- **Tracker**: indicare la natura del ticket<sub>g</sub>:
  - **Documento**: stesura di un documento. Il tipo di attività svolta dal redattore del documento viene definito durante la rendicontazione;

- **Codifica:** stesura di codice<sub>g</sub>;
- **Verifica:** macro-attività di verifica sul prodotto dei sotto-processi<sub>g</sub>.
- **Oggetto:** descrizione breve e significativa;
- **Descrizione:** descrizione comprensibile e con riferimenti esterni mediante link<sub>g</sub> se necessario;
- **Stato:** Plan;
- **Attività principale:** se si vuole creare una **sotto-attività** indicare l'id del ticket<sub>g</sub> padre;
- **Categoria:** PDCA, solo se il ticket<sub>g</sub> viene creato dal *Responsabile di Progetto*;
- **Assegnato a:** inserire il nome del responsabile;
- **Osservatori:** aggiungere eventuali collaboratori.

#### 6.7.2.2 Ticket<sub>g</sub> di realizzazione e controllo

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

### 6.7.2.3 Ticket<sub>g</sub> di verifica

Un *Verificatore* per creare un *ticket<sub>g</sub> di verifica* deve:

1. Assicurarsi che esista all'interno del progetto<sub>g</sub> l'attività *Verifica*. Su tale attività vi devono essere due sotto-attività: “Verifica - realizzazione”, “Verifica - approvazione”.  
Tutti i ticket<sub>g</sub> creati devono essere sotto-attività di: “Verifica - realizzazione”;
2. Creare quindi il ticket<sub>g</sub> 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<sub>g</sub> devono essere figli del ticket<sub>g</sub> “Verifica - realizzazione” del progetto<sub>g</sub> su cui si sta eseguendo la verifica;
  - **Assegnato a**: inserire il nome del responsabile del progetto<sub>g</sub> padre (es. responsabile delle *Norme di Progetto*).

Tutti i campi non segnalati sono da lasciare vuoti. Sarà poi compito del responsabile del progetto<sub>g</sub> 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*.

#### 6.7.2.4 Dipendenze temporali

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

- Andare su **segnalazioni**;
- Aprire il link<sub>g</sub> 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.

### 6.7.3 Aggiornamento ticket

Esistendo due tipologie di  $\text{ticket}_g$ , viene qui definito la procedura per effettuare l'aggiornamento di entrambe.

#### 6.7.3.1 Ticket<sub>o</sub> di pianificazione

- Andare sul menù **Segnalazioni**;
- Selezionare il ticket<sub>g</sub> di interesse;
- Cliccare il link<sub>g</sub> **Aggiorna**;
- Commentare ciò che si è fatto sulla form **Note**;
- Cambiare lo stato del ticket<sub>g</sub> secondo la seguente logica:
  - **Do**: quando un ticket<sub>g</sub> è in questo stato indica che una o più persone stanno lavorando su tale attività;
  - **Check**: quando un ticket<sub>g</sub> 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<sub>g</sub> di pianificazione padre.

### 6.7.3.2 Ticket<sub>g</sub> di realizzazione e controllo

- Andare sul menù **Segnalazioni**;
- Selezionare il ticket<sub>g</sub> di interesse;
- Cliccare il link<sub>g</sub> **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<sub>g</sub> secondo la seguente logica:
  - **In Progress**: quando un ticket<sub>g</sub> 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<sub>g</sub> di pianificazione padre secondo tali principi:
  - Ticket<sub>g</sub> figlio passa da New a In Progress: il ticket<sub>g</sub> padre passa da Plan a Do, o da Do a Check;
  - Ticket<sub>g</sub> figlio passa a Closed: il ticket<sub>g</sub> padre deve essere in Do o Check;
  - Tutti i ticket<sub>g</sub> figli vengono chiusi: il ticket<sub>g</sub> padre passa ad Act.

### 6.7.3.3 Ticket<sub>g</sub> di verifica

- Andare sul menù **Segnalazioni**;
- Selezionare il ticket<sub>g</sub> di interesse;
- Cliccare il link<sub>g</sub> **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<sub>g</sub> secondo la seguente logica:
  - **In Progress**: quando un ticket<sub>g</sub> 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<sub>g</sub> di pianificazione padre secondo tali principi:
  - Ticket<sub>g</sub> figlio passa da New a In Progress: il ticket<sub>g</sub> padre passa da Plan a Do, o da Do a Check;
  - Ticket<sub>g</sub> figlio passa a Closed: il ticket<sub>g</sub> padre deve essere in Do o Check;
  - Tutti i ticket<sub>g</sub> figli vengono chiusi: il ticket<sub>g</sub> padre passa ad Act.

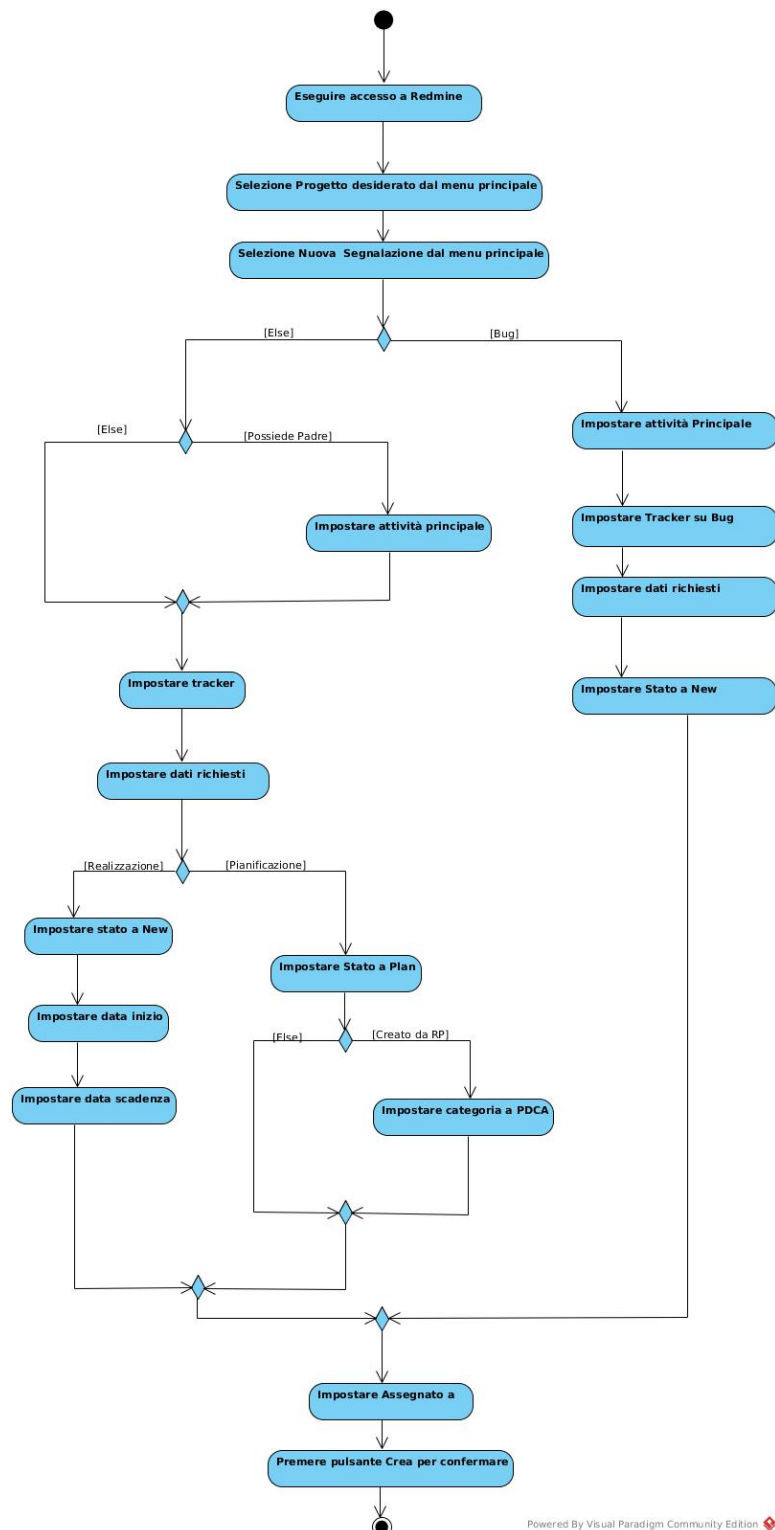


Fig 1: Diagramma attività - Creazione nuovo ticket

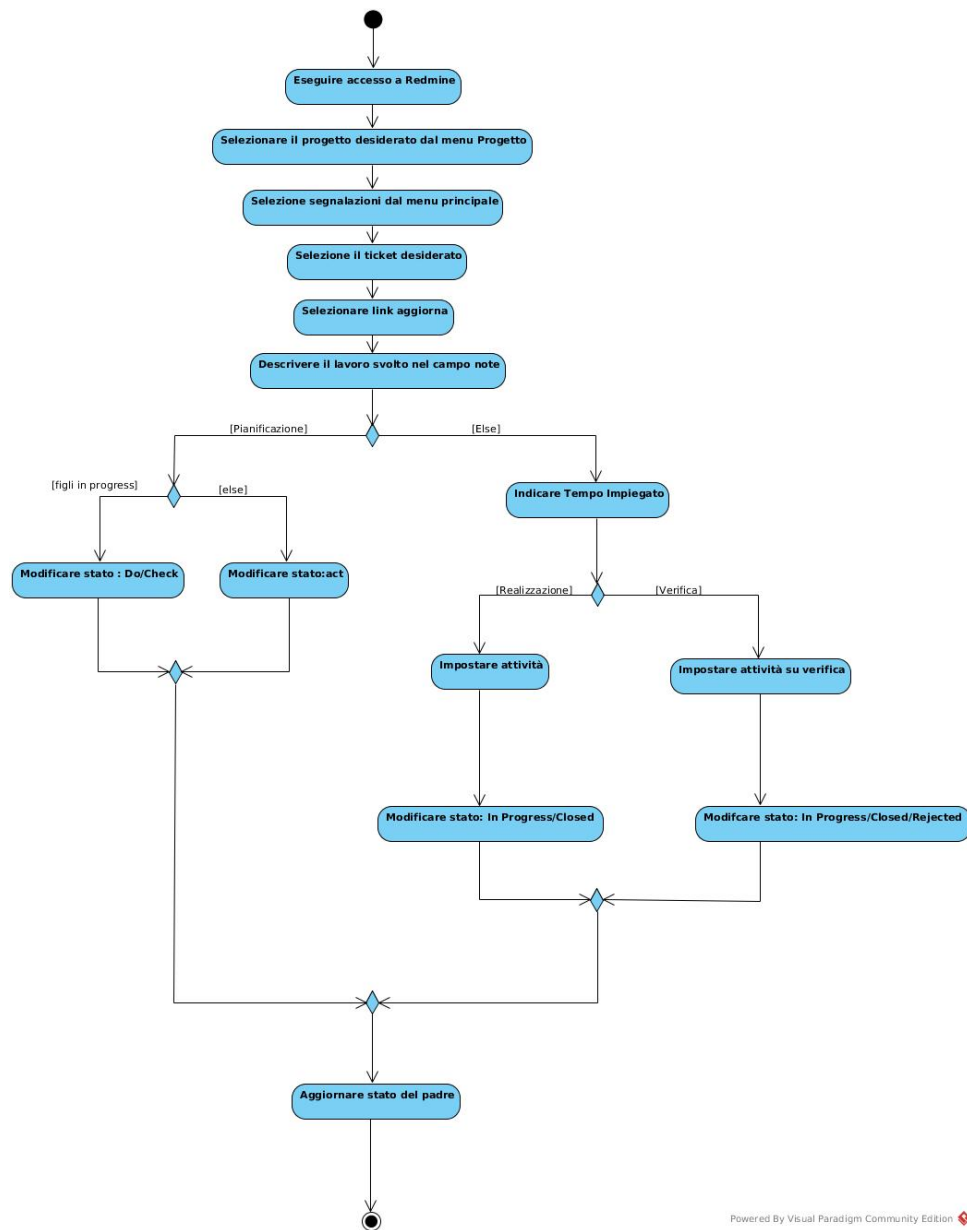


Fig 2: Diagramma attività -Aggiornamento ticket


#### 6.7.4 Consigli di utilizzo

Per avere una immediata visualizzazione dei ticket<sub>g</sub> assegnati, è consigliato personalizzare la pagina personale:

- Andare alla **Pagina personale**;
- Cliccare il link<sub>g</sub> **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**.



## 6.8 LateTack



# LateButSafe

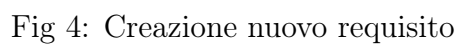
## Analisi dei Requisiti

### 6.8.1 Aggiunta nuovo requisito

---

Università degli studi di Padova - 2014/2015







Questo documento è distribuito sotto licenza [Creative Commons Attribution-NonCommercial-ShareAlike 3.0 Unported License](https://creativecommons.org/licenses/by-nc-sa/3.0/).