

# Premi

---



*Dazzle Works*

## Norme di Progetto

Versione	1.0.0
Redattori	Bogdan Suierica Emanuele Crespan
Verificatori	Matteo Agostinetto
Responsabili	Valerio Burlin
Uso	Interno
Lista di distribuzione	<i>Dazzle Works</i> Prof. Vardanega Tullio, Dr. Cardin Riccardo

Documento contenente l'insieme di norme stabilite dal gruppo *Dazzle Works* per la realizzazione di Premi.



## Indice

<b>1</b>	<b>Introduzione</b>	<b>1</b>
1.1	Scopo del Documento . . . . .	1
1.2	Scopo del Progetto . . . . .	1
1.3	Glossario . . . . .	1
1.4	Riferimenti . . . . .	1
1.4.1	Normativi . . . . .	1
<b>2</b>	<b>Collaborazione</b>	<b>2</b>
2.1	Comunicazioni . . . . .	2
2.1.1	Comunicazioni interne . . . . .	2
2.1.2	Comunicazioni esterne . . . . .	2
2.2	Composizione email . . . . .	2
2.2.1	Mittente . . . . .	2
2.2.2	Destinatario . . . . .	2
2.2.3	Oggetto . . . . .	3
2.2.4	Corpo . . . . .	3
2.2.5	Allegati . . . . .	3
2.3	Riunioni . . . . .	3
2.3.1	Interne . . . . .	3
2.3.2	Esterne . . . . .	3
2.4	Verbale . . . . .	3
2.5	Repository . . . . .	4
2.5.1	Utilizzo del servizio . . . . .	4
2.5.2	Struttura . . . . .	4
<b>3</b>	<b>Ruoli di Progetto</b>	<b>5</b>
3.1	Responsabile di Progetto . . . . .	5
3.2	Amministratore . . . . .	5
3.3	Analista . . . . .	6
3.4	Progettista . . . . .	6
3.5	Verificatore . . . . .	6
3.6	Programmatore . . . . .	6
<b>4</b>	<b>Documenti</b>	<b>7</b>
4.1	Template . . . . .	7
4.2	Versionamento . . . . .	7
4.3	Struttura dei documenti . . . . .	7
4.3.1	Prima pagina . . . . .	7
4.3.2	Diario delle modifiche . . . . .	8
4.3.3	Indice . . . . .	8
4.3.4	Formattazione delle pagine . . . . .	8
4.4	Classificazione documenti . . . . .	8
4.4.1	Documenti informali . . . . .	8
4.4.2	Documenti formali . . . . .	8
4.5	Norme tipografiche . . . . .	8
4.5.1	Punteggiatura . . . . .	9
4.5.2	Stile del testo . . . . .	9
4.5.3	Composizione del testo . . . . .	9
4.5.4	Formati ricorrenti . . . . .	9
4.5.5	Sigle . . . . .	10
4.6	Componenti grafiche . . . . .	10
4.6.1	Immagini . . . . .	10
4.6.2	Tabelle . . . . .	10



4.7	Procedure di avanzamento di un documento . . . . .	11
<b>5</b>	<b>Ambiente Lavoro</b>	<b>12</b>
5.1	Sistema operativo . . . . .	12
5.2	Coordinamento . . . . .	12
5.2.1	Repository Github . . . . .	12
5.2.2	Google Drive . . . . .	12
5.2.3	Teamwork e Facebook . . . . .	12
5.3	Ambiente documentale . . . . .	12
5.3.1	Stesura documenti . . . . .	12
5.3.2	Script . . . . .	13
5.3.3	Pianificazione delle attività . . . . .	13
5.4	Ambiente di sviluppo . . . . .	13
5.5	Diagrammi UML . . . . .	13
5.6	Verifica . . . . .	13
5.6.1	Verifica dei documenti . . . . .	13
5.7	Verifica dei diagrammi . . . . .	14
<b>6</b>	<b>Protocolli e Gestione di Progetto</b>	<b>15</b>
6.1	Creazione e gestione dei tasks . . . . .	15
6.2	Creazione delle milestone . . . . .	15
6.3	Esecuzione dei compiti . . . . .	16
6.4	Chiusura della milestone . . . . .	16
<b>7</b>	<b>Analisi Requisiti</b>	<b>17</b>
7.1	Requisiti . . . . .	17
7.2	Casi d'uso . . . . .	17
7.3	Tracciamento . . . . .	17
<b>8</b>	<b>Progettazione</b>	<b>18</b>
8.1	Diagrammi . . . . .	18
8.2	Design Pattern . . . . .	18
8.3	Classi di verifica . . . . .	18
8.4	Stile di progettazione . . . . .	18
<b>9</b>	<b>Codifica e Convenzioni</b>	<b>19</b>
9.1	Linguaggi di codifica . . . . .	19
9.2	Convenzioni di codifica . . . . .	19
9.3	Nomenclatura . . . . .	19
9.4	Intestazione di un file Javascript . . . . .	19
9.5	Commenti . . . . .	20



## Elenco delle tabelle



## Elenco delle figure

1	Procedure di avanzamento di un documento . . . . .	11
2	Creazione task list . . . . .	15
3	Creazione task . . . . .	15



# 1 Introduzione

## 1.1 Scopo del Documento

Questo documento ha lo scopo di definire le norme che tutti i membri del gruppo *DazzleWorks* dovranno seguire e rispettare durante lo svolgimento del progetto **Premi**. Tutti i membri sono tenuti a leggere il documento e a seguire le norme per garantire uniformità e una migliore efficacia ed efficienza delle attività. I membri del gruppo potranno contattare l'*Amministratore di Progetto* per eventuali suggerimenti, consigli, opinioni riguardanti le norme di progetto. L'*Amministratore di Progetto*, dopo essersi consultato con gli altri membri del gruppo, avrà la responsabilità di accettare o rifiutare i suggerimenti proposti. Il documento, in particolare, pone l'accento sui seguenti punti:

- Iterazione tra i membri del gruppo e la comunicazione verso l'esterno;
- Le modalità di accesso e le norme di utilizzo del repository<sub>G</sub>;
- Stesura documenti e convenzioni di scrittura utilizzate;
- Modalità di lavoro durante lo sviluppo del progetto;
- Organizzazione dell'ambiente di lavoro.

## 1.2 Scopo del Progetto

Lo scopo del progetto è la realizzazione di un software di presentazione di "slide", sviluppato in tecnologia HTML5<sub>G</sub> che funzioni sia su desktop che su dispositivo mobile. Si richiede agli sviluppatori di realizzare effetti grafici a supporto dello "storytelling"<sup>1</sup> che siano di livello comparabile con *Prezi*<sup>2</sup>. Il software dovrà coprire i due momenti fondamentali per questo tipo di attività: la creazione da parte dell'autore e la presentazione al pubblico, sia in presenza diretta che via web.

## 1.3 Glossario

Al fine di evitare ogni ambiguità e permettere al lettore una migliore comprensione dei termini tecnici, acronimi, e termini che necessitano di essere chiariti, essi sono riportati nel documento *Glossario*. Ogni occorrenza di un termine appartenente al *Glossario* è marcata da una "G" in pedice.

## 1.4 Riferimenti

### 1.4.1 Normativi

- <http://www.w3.org/standards/webdesign/htmlcss>

---

<sup>1</sup>Strumento riflessivo per la costruzione di significati interpretativi della realtà.

<sup>2</sup>È un servizio per la realizzazione di presentazioni.



## 2 Collaborazione

### 2.1 Comunicazioni

#### 2.1.1 Comunicazioni interne

Le comunicazioni interne verranno eseguite attraverso la mailing list<sub>G</sub>: [dazzleworksgroup@gmail.com](mailto:dazzleworksgroup@gmail.com).

Quando un membro del gruppo vuole inviare una email a tutti i componenti, deve inviare il messaggio dalla sua casella di posta elettronica personale verso l'indirizzo [dazzleworksgroup@gmail.com](mailto:dazzleworksgroup@gmail.com), dalla quale un inoltro automatico provvederà a mandare l'email agli altri indirizzi di posta elettronica personali dei componenti del gruppo.

I membri sono tenuti a prestare attenzione al numero di messaggi diffusi. È sempre importante non arrivare a una sovraesposizione del pubblico interno, in quanto si creerebbe solo senso di smarrimento e confusione. Questo non esclude che uno stesso messaggio non sia proposto su più mezzi di informazione, azione spesso necessaria, ma questo presuppone un intervento ponderato e non casuale.

Con lo scopo di facilitare la comunicazione tra i membri del gruppo vengono utilizzati strumenti di messaggistica istantanea e videochiamata come Skype e Google Hangout. È necessario redigere un verbale<sub>G</sub> nel caso in cui siano state prese decisioni o siano emersi dettagli inerenti allo sviluppo del progetto.

#### 2.1.2 Comunicazioni esterne

Per le comunicazioni esterne è stato creato un indirizzo di posta elettronica:

[dazzleworksgroup@gmail.com](mailto:dazzleworksgroup@gmail.com)

Tale indirizzo deve essere l'unico canale di comunicazione verso l'esterno. Sarà solo il *Responsabile di progetto* ad utilizzare l'indirizzo di posta per intrattenere le corrispondenze con i proponenti e i committenti. È compito del *Responsabile di progetto* informare i membri del gruppo delle discussioni avvenute ed eventualmente inoltrare i messaggi alla mailing list<sub>G</sub>.

### 2.2 Composizione email

In questo paragrafo viene descritta la forma che deve avere una email sia per comunicazione interna che esterna.

#### 2.2.1 Mittente

- **Interno:** indirizzo personale di chi scrive il messaggio;
- **Esterno:** l'unico indirizzo utilizzabile per comunicare verso l'esterno è [dazzleworksgroup@gmail.com](mailto:dazzleworksgroup@gmail.com) e deve essere usato esclusivamente dal *Responsabile di progetto*.

#### 2.2.2 Destinatario

- **Esterno:** l'indirizzo del destinatario può avere variazioni a seconda che si voglia comunicare con il Prof. Tullio Vardanega, il Prof. Riccardo Cardin o con i committenti del progetto;
- **Interno:** l'unico indirizzo utilizzabile è [dazzleworksgroup@gmail.com](mailto:dazzleworksgroup@gmail.com).

Le uniche eccezioni permesse sono:

- **Proposte all'Amministratore di progetto:** per eventuali proposte di cambiamento delle norme da parte di un membro del gruppo, quest'ultimo dovrà contattarlo al suo indirizzo di posta elettronica personale;
- **Comunicazione ristretta tra alcuni membri del gruppo:** in questi casi i membri del gruppo utilizzeranno i loro indirizzi personali.



### 2.2.3 Oggetto

L'oggetto deve sintetizzare il contenuto della email. Deve essere chiaro, breve e possibilmente univoco in modo da riconoscerlo dagli altri precedenti.

Nel caso si debba mandare un messaggio alla mailing list<sub>G</sub> vi è l'obbligo di aggiungere "Group:" all'inizio dell'oggetto.

### 2.2.4 Corpo

Il corpo del messaggio deve essere chiaro, diretto e avere tutte le informazioni necessarie per permettere a tutti i destinatari di capire correttamente l'argomento trattato. Per riferirsi ad alcuni componenti del gruppo o ruoli di progetto si dovrà usare la seguente sintassi: "**@Cognome Nome**" o "**Nome Ruolo**". Alla fine del corpo del messaggio, il mittente dovrà sempre firmarsi con il suo nome, cognome e ruolo all'interno del gruppo.

### 2.2.5 Allegati

Viene consentito l'uso di allegati che deve essere limitato solo al caso in cui essi siano realmente necessari. È buona norma presentare gli allegati scrivendo alcune righe di spiegazione. Prestare attenzione al formato (preferibilmente in formato PDF<sub>G</sub>) del documento che si sta inviando.

## 2.3 Riunioni

Il *Responsabile di progetto* ha il compito di fissare le riunioni, sia interne che esterne. Per ogni riunione sarà necessario specificare data, ora, luogo, e l'ordine del giorno. Le informazioni sulla riunione dovranno essere rese disponibili almeno quattro giorni prima.

### 2.3.1 Interne

Sarà il *Responsabile di progetto* a convocare le riunioni generali, ovvero le riunioni a cui devono partecipare tutti i componenti del gruppo. Le riunioni avranno una frequenza almeno quindicinale. Ogni componente del gruppo è tenuto a leggere la posta elettronica e rispondere ad eventuali richieste di riunioni interne.

Il *Responsabile di progetto* può anticipare o posticipare una riunione in base alla disponibilità data dai componenti del gruppo. È possibile richiedere una riunione generale da parte di un qualsiasi componente del gruppo, che verrà inoltrata al *Responsabile di progetto* che può decidere se accoglierla o rifiutarla.

Inoltre, è possibile e auspicabile che siano necessarie riunioni tra specifici membri del gruppo, ad esempio: in fase di analisi può essere utile che solo gli *Analisti* si incontrino tra di loro, senza il resto del gruppo. I restanti componenti del gruppo saranno comunque informati sui contenuti e le decisioni prese tramite invio di una email alla mailing list<sub>G</sub> o alla pubblicazione di un verbale<sub>G</sub> sul repository<sub>G</sub> nel caso siano state prese decisioni importanti.

### 2.3.2 Esterne

Sarà il *Responsabile di progetto* a fissare le riunioni esterne con i proponenti e/o i committenti utilizzando la casella di posta creata appositamente. Prima di contattare le parti esterne, il *Responsabile di progetto* dovrà verificare la presenza dei membri interni al gruppo.

In caso di risposta positiva si provvederà a contattare i proponenti e/o i committenti per fissare la data dell'incontro. È compito del *Responsabile di progetto* a redigere il verbale<sub>G</sub> dell'incontro avvenuto.

## 2.4 Verbale

Per verbali si intende quanto si è discusso nel corso di una riunione e le decisioni che eventualmente sono state assunte.





Nella stesura di un verbale occorre essere sintetici, essenziali ed esaustivi, cercando di rappresentare gli interventi, le osservazioni o le decisioni assunte in maniera oggettiva, senza interpretazioni personali, ambigue o fuorvianti. Il tempo dei verbi utilizzato è il presente.

I verbali dovranno essere nominati nel seguente criterio:

- Dove e quando si svolge la riunione, i partecipanti, gli assenti, il nome di chi presiede e quello del segretario verbalizzante;
- Tipo di incontro che potrà essere interno o esterno;
- Le questioni o i problemi ancora irrisolti;
- I (nuovi) argomenti posti in discussione all'ordine del giorno;
- Gli interventi più significativi rappresentati in ordine temporale;
- I suggerimenti, le proposte, le conclusioni e le decisioni assunte;
- I compiti assegnati ad ognuno con indicazione delle rispettive scadenze;
- Gli argomenti da trattare nella prossima riunione.

## 2.5 Repository

In questa sezione sono definiti gli strumenti utilizzati per la condivisione del file.

Per la gestione della documentazione e della codifica è stato creato un repository<sub>G</sub>. Il repository<sub>G</sub> è privato e quindi accessibile solo dai membri del gruppo *DazzleWorks*.

Per il repository<sub>G</sub> abbiamo scelto di usare il servizio GitHub<sub>G</sub>, il quale utilizza il sistema di versionamento Git<sub>G</sub>.

### 2.5.1 Utilizzo del servizio

È necessario eseguire le operazioni di sincronizzazione all'inizio e alla fine di ogni sessione di lavoro.

In particolare ad ogni sessione di lavoro andranno eseguite le seguenti operazioni:

- **Pull** - scaricare da la versione più aggiornata del progetto per poterci poi lavorare offline;
- **Commit** - serve a "scattare un'istantanea" della propria repository<sub>G</sub> per "posare" una pietra miliare all'interno del proprio progetto di sviluppo;
- **Push** - serve a pubblicare i risultati online, sulla piattaforma di sviluppo;
- **Merge** - permette di fondere un branch<sub>G</sub> con il repository<sub>G</sub> madre, così da implementare le modifiche apportate ai file e alle cartelle originarie.

### 2.5.2 Struttura

Tutta la documentazione si può trovare al seguente indirizzo: <https://github.com/FabioRos/Premi>. Tutti i file riguardanti la documentazione andranno messi in questo repository<sub>G</sub>.

Il repository<sub>G</sub> avrà questa struttura:

- Analisi dei Requisiti
- Piano di progetto
- Piano di qualifica
- Studio di fattibilità
- Glossario
- Lettera di presentazione
- Verbali
- Template



## 3 Ruoli di Progetto

Lo sviluppo prevede la collaborazione di individui a cui verranno assegnati ruoli diversi. Tali ruoli rappresentano figure aziendali specializzate, indispensabili per il buon esito del progetto. Ogni membro del gruppo dovrà ricoprire ogni ruolo almeno una volta. Si deve controllare attentamente che non vi siano conflitti di interesse specialmente nelle attività di approvazione e verifica. Per garantire che la rotazione dei ruoli non provochi conflitti è necessario che le attività vengano pianificate con attenzione e che i membri interessati rispettino il loro ruolo e compito assegnato. Sarà il compito del *Verificatore* a controllare che tutte le condizioni sopra indicate siano rispettate. Se il *Verificatore* troverà delle incongruenze con quanto menzionato sopra avrà il compito di avvisare il *Responsabile di Progetto* che avrà la responsabilità di risolvere la questione.

Si descrivano i diversi ruoli di progetto con le relative responsabilità e le modalità operative.

### 3.1 Responsabile di Progetto

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

- 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:

- Assicurarsi che le attività di verifica e validazione vengano svolte sistematicamente seguendo le *Norme di Progetto*;
- Garantire che vengano rispettati i ruoli e le competenze assegnate nel *Piano di Progetto*;
- Garantire che non vi siano conflitti tra *Verificatori* e *Redattori*;
- Gestire la creazione e l'assegnazione dei ticket e di assegnarli ad ogni membro del gruppo.

### 3.2 Amministratore

L'*Amministratore di Progetto* è il responsabile dell'efficienza e del rendimento dell'ambiente di lavoro. Le mansioni che gli competono sono le seguenti:

- Aiuta a rendere più agevole le cose che si devono fare per norma;
- Ricerca di strumenti in modo da automatizzare il maggior numero di compiti;
- Fornire procedure e strumenti per il controllo e la segnalazione del controllo qualità;
- Gestire l'archiviazione e il versionamento della documentazione;
- Controllare le versioni dei prodotti e gestire le loro configurazioni.

Redige le *Norme di Progetto*, dove spiega e norma l'utilizzo degli strumenti, e redige la sezione del *Piano di Qualifica* dove vengono descritti strumenti e metodi di verifica.



### 3.3 Analista

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

- Comprendere appieno la natura e la complessità del problema;
- Produrre una specifica di progetto motivata in ogni suo punto e comprensibile a tutti li interessati.

Redige lo *Studio di Fattibilità*, l'*Analisi dei Requisiti* e partecipa alla redazione del *Piano di Qualifica* in quanto conosce l'ambito di progetto.

### 3.4 Progettista

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

- Produrre una soluzione comprensibile, attuabile 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 relative alle metriche di verifica della programmazione del Piano di Qualifica.

### 3.5 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 non siano stati introdotti errori lungo il percorso;
- 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.

### 3.6 Programmatore

Il *Programmatore* è responsabile delle attività di codifica e delle componenti di ausilio necessarie per l'esecuzione delle prove di verifica e validazione. Le mansioni che gli competono sono:

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

Redige il *Manuale Utente*.



## 4 Documenti

In questo capitolo si descrivono i vari standard adottati da *DazzleWorks* nella stesura, verifica e approvazione della documentazione da produrre.

### 4.1 Template

Per semplificare la redazione dei documenti è stato creato un template  $\text{\LaTeX}$  contenente tutte le impostazioni per l'aspetto grafico.

Ogni documento dovrà essere realizzato con il template  $\text{\LaTeX}$  presente nel repository  $G$ . Inoltre è stata scritta una piccola guida su come usare al meglio il template.

### 4.2 Versionamento

Per ogni documento è obbligatorio specificare la versione. Un numero di versione deve essere nella seguente forma:

$$X.Y.Z$$

dove  $X$ ,  $Y$ ,  $Z$  sono interi non negativi e non devono contenere zeri iniziali.  $X$  è la versione *Major*,  $Y$  è la versione *Minor*, e  $Z$  è la versione *Patch*.

- $X$ : la versione Major ( $X.y.z \mid X > 0$ ) identifica la versione di rilascio. Deve essere incrementata se è stata introdotta qualsiasi modifica non retrocompatibile. Le versioni Patch e Minor devono essere reimpostate a 0 quando la versione Major è incrementata.
- $Y$ : la versione Minor ( $x.Y.z \mid x > 0$ ) deve essere incrementata se è stata introdotta una nuova funzionalità. La versione Patch deve essere reimpostata a 0 quando la versione Minor è incrementata.
- $Z$ : la versione Patch ( $x.y.Z \mid x > 0$ ) deve essere incrementata solo se sono state introdotte correzioni retrocompatibili di bug. Una correzione di un bug è definita come una modifica interna che corregge un comportamento errato.

Una volta che un pacchetto versionato è stato rilasciato, i contenuti di quella versione **non devono** essere modificati. Qualsiasi modifica **deve** essere rilasciata come una nuova versione.

### 4.3 Struttura dei documenti

#### 4.3.1 Prima pagina

Ogni documento deve avere in prima pagina le seguenti informazioni:

- Nome del progetto;
- Logo del gruppo;
- Nome del gruppo;
- Nome del documento;
- La versione del documento;
- Cognome e nome dei redattori del documento;
- Cognome e nome dei verificatori del documento;
- Cognome e nome di chi ha approvato il documento;
- Tipo d'uso del documento;
- Lista di distribuzione del documento;
- Breve descrizione del documento.



### 4.3.2 Diario delle modifiche

Nella seconda pagina di ogni documento deve essere presente il diario delle modifiche. Ogni riga del diario delle modifiche contiene:

- Una breve descrizione sulle modifiche effettuate;
- Cognome e nome di chi ha effettuato la modifica;
- Data della modifica;
- Versione del documento dopo la modifica.

### 4.3.3 Indice

In ogni documento deve essere presente un indice delle sezioni.

Nel caso in cui il documento contenga immagini e/o tabelle devono essere presenti anche i relativi indici.

### 4.3.4 Formattazione delle pagine

L'intestazione di ogni pagina contiene:

- Logo del gruppo;
- La sezione corrente all'interno del documento.

A piè di ogni pagina è presente:

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

## 4.4 Classificazione documenti

### 4.4.1 Documenti informali

Si definiscono documenti informali tutti i documenti in fase di sviluppo e devono ancora essere approvati dal *Responsabile di Progetto*. I documenti sono da considerarsi esclusivamente per uso interno e non potranno essere divulgati prima di essere stati verificati ed approvati.

### 4.4.2 Documenti formali

Si definiscono documenti formali tutti i documenti che sono stati approvati dal *Responsabile di Progetto* e quindi sono pronti ad essere diffusi a terze parti.

I documenti pronti per il rilascio dovranno essere rinominati osservando le seguenti regole:

- La prima lettera di ogni parola, che non sia una preposizione, deve essere maiuscola;
- Gli spazi devono essere sostituiti con il carattere "\_" (underscore);
- Tutte le parole devono essere prive di accenti;
- La versione del documento verrà aggiunta dopo il nome, e sarà preceduta dal carattere "-"(trattino) e da una "v".

## 4.5 Norme tipografiche

Questa sezione racchiude tutte le informazioni riguardanti l'ortografia, la tipografia e l'assunzione di uno stile uniforme in tutti i documenti allo scopo di evitare incoerenze tra le diverse parti del documento.



#### 4.5.1 Punteggiatura

- Punteggiatura: qualsiasi carattere di punteggiatura non può seguire un carattere di spazio;
- Lettere maiuscole: l'uso delle maiuscole è obbligatorio in una serie di casi:
  - All'inizio di testo o di una sua parte (capitolo, paragrafo, ecc.);
  - Dopo il punto, il punto esclamativo, il punto interrogativo;
  - All'inizio di ogni elemento di un elenco puntato;
  - Per i ruoli, di progetto, i nomi dei documenti, le fasi di progetto, revisioni di progetto oltre che dove previsto dalla lingua italiana.
- Parentesi: il testo racchiuso dentro le parentesi non deve mai iniziare per un carattere di spazio e non deve mai terminare con un carattere di spazio o punteggiatura.

#### 4.5.2 Stile del testo

- Maiuscolo: l'uso delle maiuscole è limitato alla trascrizione degli acronimi;
- Grassetto: l'uso del grassetto deve essere utilizzato nei seguenti casi:
  - Elenchi puntati: per evidenziare l'oggetto trattato;
  - Altri casi: è possibile utilizzare il grassetto per evidenziare parole chiave.
- Corsivo: l'uso del corsivo deve essere utilizzato nei seguenti casi:
  - Citazioni: ogni citazione deve essere scritta in corsivo;
  - Documenti: ogni riferimento ad un documento deve essere scritto in corsivo (esempio: *Glossario*);
  - Ruoli: ogni riferimento a figure particolari deve essere scritto in corsivo (esempio: *Verificatore*);
  - Altri casi: è possibile utilizzare il corsivo per evidenziare parole particolarmente significative.
- $\text{\LaTeX}$ : ogni riferimento a  $\text{\LaTeX}$  va scritto tramite il comando `\LaTeX`.

#### 4.5.3 Composizione del testo

- Elenchi puntati: per redigere un elenco si qualificano gli elementi e si introducono mediante i due punti. Dopo i due punti si deve andare a capo. Ogni elemento dell'elenco termina con un punto e virgola, eccetto l'ultimo che termina con un punto semplice;
- Pedice "G": il pedice "G" viene utilizzato in presenza di termini presenti nel *Glossario*;
- Note a piè di pagina: ogni nota deve iniziare con la lettera della prima parola maiuscola e non deve essere preceduta da alcun carattere di spazio. Ogni nota deve terminare con un punto.

#### 4.5.4 Formati ricorrenti

- Nomi propri: l'utilizzo dei nomi propri deve seguire la seguente forma "Cognome Nome";
- Percorsi: per gli indirizzi email e gli indirizzi web deve essere usato esclusivamente il comando `\url`;
- Date: ogni data deve seguire lo standard internazionale per date ed orari ISO 8601:

AAAA.MM.GG

dove:



- AAAA: rappresenta il formato dell'anno scritto con quattro cifre;
- MM: rappresenta il formato del mese scritto con due cifre;
- GG: rappresenta il formato del giorno scritto con due cifre.
- Riferimenti ai documenti: ci si riferirà ai vari documenti scrivendo in corsivo il nome del documento e mettendo una lettera maiuscolo per ogni parola che non sia un articolo. Nel caso in cui ci si deve riferire ad una versione specifica del documento, essa andrà indicata alla fine del nome del documento (esempio: *Norme di Progetto v1.0.0*);
- Nome del gruppo: ci si riferirà al gruppo solo come "*Dazzle Works*" con il nome del gruppo in corsivo. Per una scrittura corretta del nome è stato creato il comando `\LATEX\GRUPPO`;
- Nome del progetto: ci si riferirà al nome del progetto solo come "**PREMI**". Per una scrittura corretta del nome è stato creato il comando `\LATEX\PROGETTO`.

#### 4.5.5 Sigle

Le sigle potranno essere utilizzate esclusivamente all'interno delle tabelle e diagrammi. Sono presenti le seguente sigle:

- Documenti:
  - NdP = *Norme di Progetto*;
  - AdR = *Analisi dei Requisiti*;
  - PdP = *Piano di Progetto*;
  - PdQ = *Piano di Qualifica*;
  - SdF = *Studio di Fattibilità*.
- Revisioni:
  - RR = *Revisione dei Requisiti*;
  - RP = *Revisione di Progettazione*;
  - RQ = *Revisione di Qualifica*;
  - RA = *Revisione di Accettazione*.

## 4.6 Componenti grafiche

### 4.6.1 Immagini

Tutte le immagini dovranno avere il formato PDF<sub>G</sub> oppure PNG. In modo da garantire una maggiore qualità dell'immagine in caso di ridimensionamento è consigliato usare il formato PDF<sub>G</sub> vettoriale. La conversione di un immagine in formato PDF<sub>G</sub> è possibile attraverso il software GIMP<sup>3</sup>, il quale è stato già usato da tutti i componenti del gruppo.

Le immagini devono essere accompagnate da una didascalia che inizia con la parola "Figura" con la prima lettera maiuscola, seguita dal numero della figura, dal carattere "due punti" e da una breve descrizione della figura.

### 4.6.2 Tabelle

Ogni tabella deve essere accompagnata da una didascalia che inizia con la parola "Tabella" con la prima lettera maiuscola, seguita dal numero della tabella, dal carattere "due punti" e da una breve descrizione non banale della tabella.

---

<sup>3</sup>GNU Image Manipulation Program è un programma per la creazione e modifica di immagini digitali.



## 4.7 Procedure di avanzamento di un documento

Quando il redattore ritiene di aver finito la stesura di un documento si devono seguire questi passi esattamente in quest'ordine:

1. Il redattore ha il compito di contattare il *Responsabile di Progetto* per informarlo della terminazione della stesura del documento;
2. Il *Responsabile di Progetto* provvederà a contattare ed assegnare la verifica del documento ad un verificatore disponibile, che non sia in conflitto di interessi;
3. Il *Verificatore* provvederà alla verifica del documento;
4. Nel caso in cui vengano trovati errori nel documento, il *Verificatore* provvederà alla creazione di un ticket di correzione e lo assegnerà al redattore, il quale correggerà gli errori trovati e infine si tornerà di nuovo al passo uno;
5. Se non sono stati trovati errori, il documento verrà approvato dal *Responsabile di Progetto*.

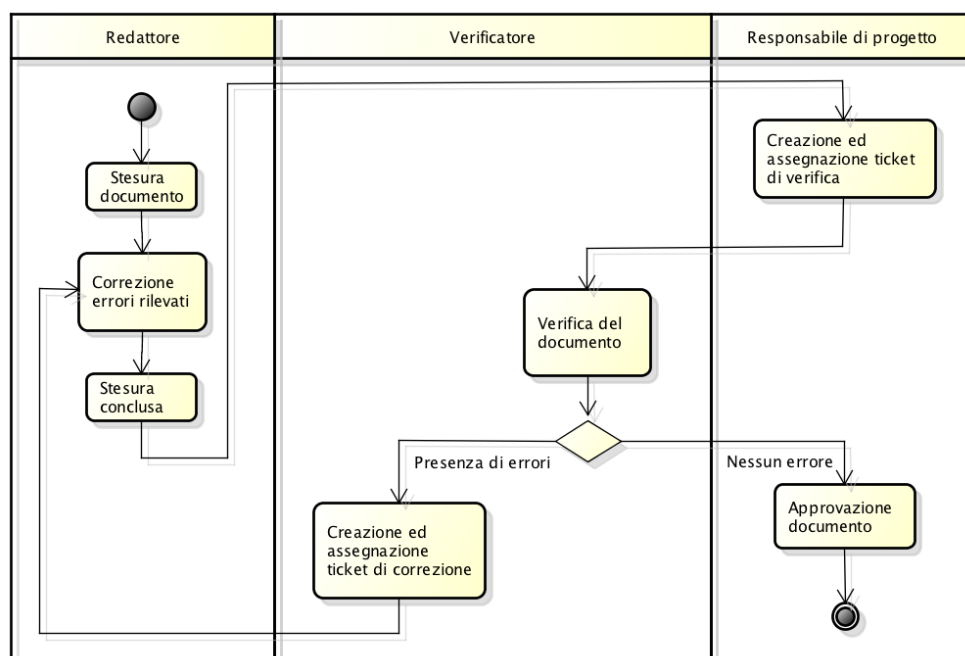


Figura 1: Procedure di avanzamento di un documento





## 5 Ambiente Lavoro

In questa sezione verrà illustrato l'ambiente di lavoro che sarà utilizzato durante lo sviluppo del progetto Premi.

### 5.1 Sistema operativo

Il sistema operativo utilizzato è lasciato a discrezione di ogni membro del gruppo. Questa scelta è dovuta principalmente al fatto che il progetto dovrà supportare più piattaforme. I membri del gruppo utilizzeranno i seguenti sistemi operativi:

- Windows 7 64 bit;
- Ubuntu 14;
- Mac OS X.

### 5.2 Coordinamento

Il coordinamento del gruppo avviene tramite:

- Repository Github;
- Google Drive;
- Teamwork;
- Facebook.

#### 5.2.1 Repository Github

Nonostante siano disponibili molti repository (Github, Mercurial, SVN) è stato scelto di utilizzare Github in quanto il servizio soddisfa pienamente le necessità di hosting e versionamento necessarie per lo sviluppo di questo progetto, inoltre diversi membri del gruppo avevano già usato tale repository.

#### 5.2.2 Google Drive

Abbiamo scelto di utilizzare questo servizio cloud per condividere tutti i documenti che vengono utilizzati frequentemente da parte dei membri del gruppo e per la semplicità di accesso dei documenti via browser.

#### 5.2.3 Teamwork e Facebook

Un' altro servizio che teamwork offre, oltre a quello di poter gestire le attività, e quello di poter indicare se una persona è assente o non reperibile in certe date agevolando la gestione delle risorse umane. Facebook viene utilizzato per inviare messaggi attraverso i membri del gruppo.

### 5.3 Ambiente documentale

#### 5.3.1 Stesura documenti

Per la stesura dei documenti verrà utilizzato il linguaggio di markup  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ . Come editor è consigliato TexMaker, il quale è disponibile per tutti i principali sistemi operativi.



### 5.3.2 Script

Per facilitare la stesura dei documenti sono stati creati alcuni script:

- Evidenziare glossario: con il comando `java glossary` verrà eseguito uno script che selezionerà nei documenti, della directory corrente, le parole contenute nella versione più recente del glossario e le evidenzierà con il simbolo  $G$ .  
"scrivo altri script quando abbiamo deciso quali fare"

### 5.3.3 Pianificazione delle attività

Per pianificare le attività verrà utilizzato il servizio Teamwork che permette, oltre a creare task ed assegnarli ai vari membri del gruppo, di creare in automatico il diagramma di Gantt.

## 5.4 Ambiente di sviluppo

"Descrizione IDE e Framework vari."

## 5.5 Diagrammi UML

Per la realizzazione dei diagrammi UML è stato scelto Astah Professional Edition. E' possibile ricevere gratuitamente la licenza della versione professional inviando una richiesta al sito <http://astah.net/student-license-request>.

## 5.6 Verifica

### 5.6.1 Verifica dei documenti

La verifica dei documenti verrà eseguita ogni volta che è stata effettuata una modifica ad un documento e debba essere approvato. Per una corretta verifica di un documento vanno seguite le seguenti pratiche:

- 1) Controllo tipografico: tramite l'utilizzo di Texmaker e di aspell verranno trovati errori tipografici presenti nel documento;
- 2) Controllo lessicale: il Verificatore dovrà controllare che il documento non presenti errori lessicali attraverso un'attenta analisi del testo utilizzando la tecnica inspection o walkthrough;
- 3) Controllo glossario: il verificatore dovrà controllare che ogni parola, nel testo, presente nel glossario sia correttamente evidenziata;
- 4) Controllo contenuto: il Verificatore dovrà controllare che il documento contenga tutto il necessario e che sia impaginato adeguatamente;
- 5) Rispetto delle norme del progetto: il Verificatore dovrà controllare che il documento segua le norme di progetto stabilite;
- 6) Lista di controllo: si dovrà stilare una lista di errori più frequenti, per semplificare le successive verifiche dei documenti;
- 7) Rispetto indice Gulpease: il verificatore dovrà calcolare e controllare, per ogni documento, che gli indici di Gulpease risiedano nel range di valori specificato nel Piano di Qualifica, altrimenti si dovrà effettuare una ricerca walkthrough alla ricerca delle frasi troppo lunghe e complesse;
- 8) Segnalazione errori: una volta completata la verifica di un documento se sono stati riscontrati errori, il Verificatore dovrà aprire dei task per segnalarli.



## 5.7 Verifica dei diagrammi

I diagrammi devono essere verificati manualmente dal verificatore e deve controllare che aderiscano correttamente allo standard UML. In particolare deve controllare che i diagrammi di flusso siano rappresentati in maniera corretta e che i casi d'uso utilizzino correttamente le inclusioni e le estensioni.



## 6 Protocolli e Gestione di Progetto

Il servizio Teamwork fornisce un sistema per la gestione dei tasks e creazione delle milestone. Si è quindi deciso di utilizzare questo servizio.

### 6.1 Creazione e gestione dei tasks

I tasks vengono creati e gestiti quasi tutti dal responsabile di progetto. Qualora il verificatore trovasse imprecisioni od errori trovati durante la verifica avrà la possibilità di creare dei tasks per segnalare suddetti errori.

Ogni task può essere assegnato ad uno o più membri del gruppo a seconda della complessità del lavoro e alla disponibilità dei membri del gruppo; i tasks avranno le seguenti caratteristiche:  
Per creare un nuovo task bisogna:

- Andare su Tasks;
- Premere add Task list;
- Compilare i campi richiesti;
- Aggiungere una task alla task list;
- Compilare i campi richiesti.

Figura 2: Creazione task list

Figura 3: Creazione task

### 6.2 Creazione delle milestone

Il responsabile di progetto ha il compito della creazione di una milestone in occasione di ogni revisione al quale il gruppo DazzleWork ha intenzione di partecipare, più altre milestone qualora il responsabile di progetto lo ritenga necessario.

Per creare una nuova milestone bisogna:

- Andare su milestone;



- Premere add a milestone;
- Compilare i campi richiesti.

The screenshot shows a modal dialog titled "Add a new milestone" with a close button (X) in the top right corner. The dialog contains the following fields and options:

- Calendar:** A small calendar icon showing "MAY 2015" and a large number "1" for Sunday.
- Name:** A text input field with the placeholder "Give this milestone a name".
- Description:** A text area with the placeholder "Add Description...".
- When is it due?:** A date picker showing "01/03/2015".
- Who's responsible?:** A dropdown menu showing "prova Unknown (Me)" with a "Multiple People" link below it.
- Privacy:** A dropdown menu set to "Everybody on project".
- Followers:** A section stating "Nobody is following status updates on this milestone" with a "Change" link.
- Reminder?:** A checkbox labeled "(No email reminder will not be sent to the person responsible)".
- Notify now?:** A checkbox labeled "(The person responsible will not be notified)".

At the bottom, there are two buttons: "Add this milestone" (in green) and "Cancel".

### 6.3 Esecuzione dei compiti

Ogni membro del gruppo è tenuto a visionare regolarmente la presenza di tasks a lui assegnati e segnalarne la presa in consegna. Una volta che un membro porta a termine un task deve modificarne lo stato per segnalare il termine del lavoro. Se un task scade prima che sia terminato il responsabile di progetto può riaprirlo ed eventualmente assegnare altri membri al lavoro.

### 6.4 Chiusura della milestone

Una volta raggiunta la scadenza, il responsabile di progetto, deve chiudere la milestone ed eventualmente aprirne un'altra per poi ricominciare tutto il protocollo da capo.



## 7 Analisi Requisiti

E' compito degli analisti stilare il documento Analisi dei requisiti. In questo documento dovranno essere presenti tutti i requisiti ed i casi d'uso emersi dall'analisi del capitolato e dalle riunioni con il proponente.

### 7.1 Requisiti

I requisiti dovranno essere classificati per tipo e priorità, utilizzando la seguente notazione:

R{importanza}{tipo}{codice}

- Importanza può assumere i seguenti valori:
  - 0 : Requisito obbligatorio;
  - 1 : Requisito desiderabile;
  - 2 : Requisito opzionale.
- Tipo può assumere i seguenti valori:
  - F : Requisito funzionale;
  - Q : Requisito di qualità
  - P : Requisito di prestazione;
  - V : Requisito di vincolo.
- Codice rappresenta il codice univoco di ogni requisito in forma gerarchica.

Ogni requisito dovrà essere inserito (nel documento Analisi dei Requisiti) in una tabella contenente l'identificativo, una breve descrizione e la fonte.

### 7.2 Casi d'uso

Dopo la stesura dei requisiti è sempre compito degli analisti analizzare i casi d'uso (abbreviati con UC, use case). Per ogni caso d'uso è richiesto:

- Codice identificativo: nel formato UCxx, dove xx indica un numero identificativo del caso d'uso;
- Titolo: titolo sintetico del caso d'uso;
- Descrizione: una breve descrizione chiara e dettagliata del caso d'uso;
- Diagramma: dovrà essere usato UML 2.4 per la creazione dei diagrammi dei casi d'uso;
- Attori: gli attori coinvolti nel caso d'uso (principali e secondari);
- Precondizione: la precondizione del requisito;
- Postcondizione: la postcondizione del requisito.

### 7.3 Tracciamento

Per il tracciamento dei requisiti è stato creato un database, accessibile via web, che associa ad ogni caso d'uso i corrispettivi requisiti e viceversa.



## 8 Progettazione

Dopo la fase di Analisi si passa alla fase di Progettazione. Durante la fase di Progettazione i progettisti dovranno seguire le seguenti regole.

### 8.1 Diagrammi

Dovrà essere utilizzato il linguaggio UML per la realizzazione dei seguenti diagrammi:

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

### 8.2 Design Pattern

I progettisti dovranno utilizzare il design pattern(g) che ritengono più adatto al contesto per rendere l'applicazione più efficiente possibile.

### 8.3 Classi di verifica

Andranno create delle classi di verifica per testare che i vari componenti abbiano un comportamento corretto.

### 8.4 Stile di progettazione

Durante la fase di Progettazione bisognerà fare attenzione a:

- **Ricorsione:** non dovrà essere utilizzata la ricorsione a meno che non sia necessaria, in caso dovrà essere fornita una dimostrazione induttiva sulla correttezza del metodo in questione;
- **Annidamento di cicli:** all'interno di un metodo non dovranno esserci cicli annidati con una profondità maggiore a cinque.



## 9 Codifica e Convenzioni

### 9.1 Linguaggi di codifica

Dopo un'analisi del capitolato d'appalto e dei requisiti si è deciso che per lo sviluppo del software richiesto si utilizzeranno i linguaggi HTML e Javascript.

### 9.2 Convenzioni di codifica

Di seguito è riportato l'insieme di norme e convenzioni che il gruppo dovrà seguire nella scrittura e documentazione del codice. L'unica lingua ammessa per i nomi di variabili, classi, metodi e commenti è l'inglese.

### 9.3 Nomenclatura

Per l'assegnazione di nomi a variabili, metodi e costanti andranno seguite le seguenti regole:

- Funzioni: va utilizzata la notazione mixed case, con la prima lettera minuscola;
- Variabili: va utilizzata la notazione mixed case, con la prima lettera minuscola;
- Costanti: va scritto il nome interamente in maiuscolo, separando le varie parole con il carattere "\_" (underscore).

### 9.4 Intestazione di un file Javascript

/\*

File  
Autore  
Data  
Descrizione

Modifiche:

Versione	-	Data	-	Programmatore	-	Modifica	-	Descrizione
x.y.z	-	aaaa-mm-gg	-	Nome Cognome	-	Funzione	-	Descrizione modifica

\*/

- File : nome del file;
- Autore : nome e cognome del creatore del file;
- Data : data di creazione del file nel formato aaaa-mm-gg;
- Descrizione : poche righe di descrizione delle funzionalità contenute nel file;
- Cambiamenti : tabella dello stato di avanzamento del file, contenente tutte le modifiche effettuate :
  - Versione : versione una volta effettuata la modifica;
  - Data : data della modifica;
  - Programmatore : nome e cognome del programmatore che ha effettuato la modifica;
  - Modifica : segnatura della funzione a cui è stata apportata una modifica;
  - Descrizione : breve descrizione della modifica effettuata.





## 9.5 Commenti

Prima di ogni funzione dovrà essere presente un commento con la seguente forma:

```
/*
```

Descrizione della funzione

Descrizione dei parametri

Descrizione del tipo di ritorno

```
*/
```

Ogni variabile di particolare importanza dovrà essere fornita di commento che ne spieghi il suo scopo e funzionamento.