

# Norme di Progetto

*Gruppo LaTeXBiscotti — Progetto UMAP*

Informazioni sul documento	
Versione	3.0.0
Redazione	Federica Speggiorin
Verifica	Pietro Marchetto
Approvazione	Marco Baggio
Uso	Interno
Distribuzione	Prof. Tullio Vardanega
	Prof. Riccardo Cardin
	Gruppo LaTeXBiscotti

## Descrizione

Questo documento descrive le regole, gli strumenti e le procedure adottate dal gruppo LaTeXBiscotti per la realizzazione del progetto UMAP.



## Diario delle Modifiche

Versione	Data	Persone coinvolte	Descrizione
3.0.0	2016-05-15	Marco Baggio (Responsabile)	Approvazione documento.
2.1.0	2016-04-23	Pietro Marchetto (Verifica)	Verifica documento.
2.0.2	2016-04-22	Federica Speggiorin (Amministratore)	Inserimento sezione §6.8.
2.0.1	2016-04-21	Federica Speggiorin (Amministratore)	Sposta sottosezione §3.1 Studio di Fattibilità e Analisi dei Rischi in §2 Fornitura. Eliminato Appendice A.
2.0.0	2016-04-11	Simone Garbin (Responsabile)	Approvazione documento.
1.2.0	2016-03-16	Giovanni Rodighiero (Verificatore)	Verifica documento.
1.1.7	2016-03-14	Marco Baggio (Amministratore)	Completamento sezione §6.
1.1.6	2016-03-13	Marco Baggio (Amministratore)	Inserimento sezione §9.1.1.1.
1.1.5	2016-03-03	Marco Baggio (Amministratore)	Sistemazione riferimenti alle sezioni in seguito alla riorganizzazione della struttura del documento.
1.1.4	2016-02-26	Marco Baggio (Amministratore)	Riorganizzazione struttura di §7 (eliminata titolo sezione iniziale ridondante perchè uguale al titolo della parte corrispondente).
1.1.3	2016-02-26	Marco Baggio (Amministratore)	Aggiunta item lista in §3.1.1, ampliamento sostanzioso paragrafo §3.2.1 e relativi sotto paragrafi, ampliamento sostanzioso paragrafo §3.2.2 e relativi sotto paragrafi, aggiunti paragrafi §3.2.2.1, §3.2.2.2, inseriti due paragrafi (§3.3.3.1, §3.3.3.2) per strutturare meglio il paragrafo §3.3.3, aggiunta sezione §5 per catalogare meglio gli argomenti ivi trattati e sezione §6 (Verifica).
1.1.2	2016-02-26	Marco Baggio (Amministratore)	Aggiunta sezione 2: “Fornitura”, con lo spostamento dei paragrafi §6.1.1.1 e “Svolgimento Riunione”.
1.1.1	2016-02-25	Marco Baggio (Amministratore)	Eliminata ridondanza del termine “processo” dai titoli dei sottoparagrafi. Modificata struttura generale con l’aggiunta delle “parti”, in modo da dividere secondo una logica migliore il contenuto del documento. Modificata struttura dei paragrafi di §3.



1.1.0	2016-02-06	Federica Speggiorin (Verificatore)	Verifica documento.
1.0.1	2016-02-03	Filippo Todescato (Amministratore)	Inserimento sezione §9.3.2, riguardante il software di tracciamento.
1.0.0	2016-01-21	Pietro Marchetto (Responsabile)	Approvazione documento.
0.3.0	2016-01-08	Federica Speggiorin (Verificatore)	Verifica documento.
0.2.1	2016-01-07	Andrea Barcaro (Amministratore)	Riorganizzazione indice.
0.2.0	2015-12-31	Federica Speggiorin (Verificatore)	Verifica documento.
0.1.3	2015-12-28	Andrea Barcaro (Amministratore)	Aggiunto appendice A.
0.1.2	2015-12-27	Andrea Barcaro (Amministratore)	Aggiunta Sezione segnalazione bug.
0.1.1	2015-12-27	Andrea Barcaro (Amministratore)	Aggiunta Sezione Riferimenti Normativi, modifica regola versionamento dei documenti.
0.1.0	2015-12-26	Federica Speggiorin (Verificatore)	Verifica documento.
0.0.10	2015-12-26	Andrea Barcaro (Amministratore)	Aggiunta Software scrittura documenti LaTeX.
0.0.9	2015-12-23	Andrea Barcaro (Amministratore)	Modifica diagrammi di attività.
0.0.8	2015-12-22	Andrea Barcaro (Amministratore)	Correzioni ortografiche, aggiunta software per le presentazioni.
0.0.7	2015-12-21	Andrea Barcaro (Amministratore)	Terminata sezione “Ambiente di Lavoro”, stesura sezione “Procedure a Supporto dei Processi”, aggiunta sottosezione “Repository”.
0.0.6	2015-12-20	Andrea Barcaro (Amministratore)	Continua stesura struttura sezione “Ambiente di Lavoro”, aggiunta diagrammi di attività.
0.0.5	2015-12-19	Andrea Barcaro (Amministratore)	Stesura struttura sezione “Ambiente di Lavoro”.
0.0.4	2015-12-18	Andrea Barcaro (Amministratore)	Modifica sezione “Documenti”, riorganizzazione capitoli 2 e 3 in unico capitolo “Collaborazione”.
0.0.3	2015-12-17	Andrea Barcaro (Amministratore)	Stesura sezione “Documenti”.
0.0.2	2015-12-16	Andrea Barcaro (Amministratore)	Stesura sezioni “Introduzione”, “Comunicazioni”, “Riunioni”.
0.0.1	2015-12-16	Andrea Barcaro (Amministratore)	Stesura indice delle sezioni.



## Indice

<b>1</b>	<b>Introduzione</b>	<b>6</b>
1.1	Scopo del Documento . . . . .	6
1.2	Scopo del Prodotto . . . . .	6
1.3	Glossario . . . . .	6
1.4	Riferimenti . . . . .	6
1.4.1	Informativi . . . . .	6
<b>I</b>	<b>Processi Primari</b>	<b>7</b>
<b>2</b>	<b>Fornitura</b>	<b>8</b>
2.1	Comunicazioni Esterne . . . . .	8
2.2	Riunioni Esterne . . . . .	8
2.3	Svolgimento Riunione . . . . .	8
2.4	Studio di Fattibilità e Analisi dei Rischi . . . . .	8
<b>3</b>	<b>Sviluppo</b>	<b>10</b>
3.1	Analisi dei Requisiti . . . . .	10
3.1.1	Classificazione dei Requisiti . . . . .	10
3.1.2	Casi d'Uso . . . . .	10
3.2	Progettazione . . . . .	11
3.2.1	Specifica Tecnica . . . . .	11
3.2.1.1	Tecnologie Adottate . . . . .	11
3.2.1.2	Diagrammi UML . . . . .	11
3.2.1.3	Design Pattern . . . . .	11
3.2.1.4	Tracciamento Componenti . . . . .	12
3.2.1.5	Test di Integrazione . . . . .	12
3.2.2	Definizione di Prodotto . . . . .	12
3.2.2.1	Introduzione . . . . .	12
3.2.2.2	Standard di Progetto . . . . .	12
3.2.2.3	Definizione di Classe . . . . .	12
3.2.2.4	Tracciamento delle Classi . . . . .	13
3.2.2.5	Test di Unità . . . . .	13
3.3	Codifica . . . . .	13
3.3.1	Nomi . . . . .	13
3.3.2	Ricorsione . . . . .	13
3.3.3	Documentazione del Codice . . . . .	13
3.3.3.1	Classi . . . . .	13
3.3.3.2	Metodi . . . . .	14
<b>II</b>	<b>Processi di Supporto</b>	<b>15</b>
<b>4</b>	<b>Documentazione</b>	<b>16</b>
4.1	Template . . . . .	16
4.2	Struttura del Documento . . . . .	16
4.2.1	Frontespizio . . . . .	16
4.2.2	Diario delle Modifiche . . . . .	16
4.2.3	Indici . . . . .	16
4.2.4	Formattazione delle Pagine . . . . .	16
4.3	Norme Tipografiche . . . . .	17
4.3.1	Punteggiatura e Norme Grammaticali . . . . .	17
4.3.2	Stile di Testo . . . . .	17



4.3.3	Composizione del Testo . . . . .	18
4.3.4	Formati . . . . .	18
4.3.5	Sigle . . . . .	18
4.4	Componenti Grafiche . . . . .	19
4.4.1	Tabelle . . . . .	19
4.4.2	Immagini . . . . .	19
4.5	Classificazione Documenti . . . . .	19
4.5.1	Documenti informali . . . . .	19
4.5.2	Documenti formali . . . . .	19
4.6	Glossario . . . . .	19
<b>5</b>	<b>Gestione della Configurazione</b>	<b>20</b>
5.1	Versionamento . . . . .	20
5.2	Ciclo di Vita . . . . .	20
<b>6</b>	<b>Verifica e Validazione</b>	<b>21</b>
6.1	Metriche . . . . .	21
6.2	Verifica dei processi . . . . .	21
6.3	Verifica dei Documenti . . . . .	21
6.3.1	Walkthrough . . . . .	21
6.3.2	Inspection . . . . .	21
6.4	Verifica del tracciamento dei requisiti . . . . .	21
6.5	Verifica dei diagrammi UML . . . . .	22
6.6	Verifica della progettazione . . . . .	22
6.7	Verifica del codice . . . . .	22
6.8	Validazione . . . . .	22
6.8.1	Manuali Utente . . . . .	22
6.8.2	Codice . . . . .	22
<b>III</b>	<b>Processi Organizzativi</b>	<b>23</b>
<b>7</b>	<b>Gestione di Progetto</b>	<b>24</b>
7.1	Pianificazione delle Attività . . . . .	24
7.2	Coordinazione e Controllo delle Attività . . . . .	24
7.3	Gestione e Controllo delle Risorse . . . . .	24
7.4	Analisi e Gestione dei Rischi . . . . .	24
7.5	Elaborazione dei Dati . . . . .	24
7.6	Delega . . . . .	24
7.7	Assegnazione di Attività . . . . .	24
7.8	Gestione dei Cambiamenti . . . . .	24
7.9	Ruoli di Progetto . . . . .	25
7.9.1	Amministratore . . . . .	25
7.9.2	Analista . . . . .	25
7.9.3	Progettista . . . . .	25
7.9.4	Programmatore . . . . .	25
7.9.5	Responsabile di Progetto . . . . .	25
7.9.6	Verificatore . . . . .	25
<b>8</b>	<b>Struttura Organizzativa</b>	<b>27</b>
8.1	Collaborazione . . . . .	27
8.1.1	Comunicazioni . . . . .	27
8.1.1.1	Comunicazioni Interne . . . . .	27
8.1.1.2	Composizione Post . . . . .	27
8.1.2	Riunioni di Gruppo . . . . .	28



8.1.2.1	Frequenza . . . . .	28
8.1.3	Convocazione Riunione . . . . .	28
8.1.3.1	Verbale . . . . .	29
8.1.4	<i>Repository<sub>G</sub></i> e Strumenti di Condivisione File . . . . .	29
8.1.4.1	Struttura <i>Repository<sub>G</sub></i> dei Documenti . . . . .	29
8.1.4.2	Condivisione dei File . . . . .	29
<b>9</b>	<b>Strumenti</b>	<b>30</b>
9.1	Coordinamento . . . . .	30
9.1.1	Software di Versionamento . . . . .	30
9.1.1.1	Workflow . . . . .	30
9.1.2	Condivisione dei File . . . . .	30
9.2	Software di Pianificazione e Gestione dei Compiti . . . . .	30
9.3	Strumenti per i Documenti . . . . .	31
9.3.1	L <sup>A</sup> T <sub>E</sub> X . . . . .	31
9.3.2	Mnemosyne . . . . .	31
9.3.3	Verifica . . . . .	31
9.3.4	Diagrammi UML . . . . .	32
9.3.5	Fogli di Calcolo . . . . .	32
9.3.6	Software per le Presentazioni . . . . .	32
9.4	Protocollo per lo Sviluppo dell'Applicazione . . . . .	32
9.4.1	Creare un Nuovo Progetto . . . . .	32
9.4.2	Assegnazione di un Nuovo Compito . . . . .	32
9.4.3	Modifica di un Compito . . . . .	33
9.4.4	Segnalazione ed Assegnazione di una Issue (bug) . . . . .	34
9.5	Strumenti per lo Sviluppo dell'Applicazione e per la Codifica . . . . .	35
9.5.1	Framework . . . . .	35

## Elenco delle figure

1	Diagramma attività - Svolgimento Riunione. . . . .	8
2	Diagramma attività - Creazione Nuovo Compito. . . . .	33
3	Diagramma attività - Modifica Compito. . . . .	34
4	Diagramma attività - Segnalazione e Assegnazione Bug tramite Issue. . . . .	35



# 1 Introduzione

## 1.1 Scopo del Documento

Questo documento definisce le norme che i membri del gruppo LaTeXeBiscotti adotteranno nello svolgimento del progetto “UMAP”.

Ognuno dei membri è tenuto a leggere il suddetto documento e a seguire le regole qui riportate in modo tale da migliorare l’uniformità del prodotto, incrementare la sua efficienza e ridurre al minimo il numero di errori. In particolare verranno fornite norme riguardanti:

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

## 1.2 Scopo del Prodotto

Lo scopo del progetto è la realizzazione di un *algoritmo predittivo<sub>G</sub>* in ambiente *Internet of Things<sub>G</sub>*, in grado di analizzare i dati provenienti da “oggetti” inseriti in diversi contesti e fornire delle previsioni su possibili guasti o interazioni con nuovi utenti ed identificare dei pattern di comportamento di questi ultimi, al fine di prevedere le azioni degli stessi rispetto ad altri oggetti o ad altri contesti.

## 1.3 Glossario

Per evitare tutte le possibili incomprensioni e ambiguità sul linguaggio utilizzato e per massimizzare la comprensione da parte di tutti del documento, della terminologia specifica e di quella di dominio, degli acronimi e di tutte quelle parole che necessitano chiarimento, è stato redatto un Glossario, consultabile nel documento *Glossario v3.0.0*. Tutti i termini, la cui spiegazione è presente in *Glossario v3.0.0*, sono evidenziati, a tale scopo, con una G pedice.

## 1.4 Riferimenti

### 1.4.1 Informativi

- *ISO<sub>G</sub>/IEC<sub>G</sub> 12207-1995*  
[http://www.math.unipd.it/~tullio/IS-1/2009/Approfondimenti/ISO\\_12207-1995.pdf](http://www.math.unipd.it/~tullio/IS-1/2009/Approfondimenti/ISO_12207-1995.pdf).



## Parte I

# Processi Primari



## 2 Fornitura

### 2.1 Comunicazioni Esterne

Per quanto riguarda le comunicazioni esterne, è stato creato un account Google con associata una casella di posta elettronica:

latexebiscotti@gmail.com

Tale indirizzo mail sarà l'unico canale di comunicazione tra il gruppo e l'esterno. A tutti i membri del team sarà permesso l'accesso, ma con l'obbligo di non eliminare mai alcun tipo di mail ricevuta o inviata verso l'esterno. L'unica persona a cui sarà concesso inviare mail verso l'esterno sarà il *Responsabile di Progetto*.

### 2.2 Riunioni Esterne

Le riunioni esterne coinvolgeranno i Proponenti o i Committenti e una rappresentanza del gruppo della quale dovrà far parte obbligatoriamente il *Responsabile di Progetto*. Qualunque membro potrà richiedere una riunione, anche se tale richiesta dovrà essere sempre approvata dal *Responsabile di Progetto*.

### 2.3 Svolgimento Riunione

Prima di una riunione esterna sarà scelto tra i membri del gruppo partecipanti un Segretario; egli avrà il compito di annotare ogni argomento trattato e di redigere il verbale dell'assemblea, che dovrà poi essere inserito tramite post nella categoria specifica dei verbali, come specificato in . Il Segretario deve inoltre controllare che venga seguito l'ordine del giorno in modo da non tralasciare alcun punto.

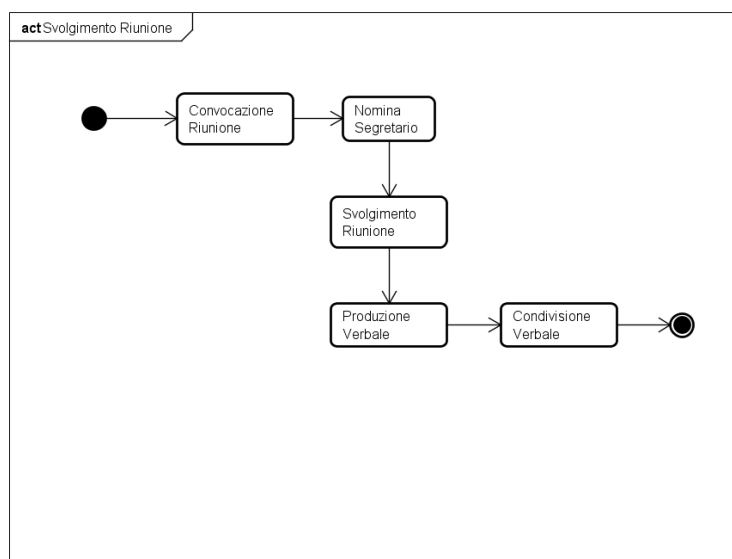


Figura 1: Diagramma attività - Svolgimento Riunione.

### 2.4 Studio di Fattibilità e Analisi dei Rischi

Una volta pubblicati i capitolati, è compito del *Responsabile di Progetto* convocare tutte le riunioni necessarie per consentire il confronto tra tutti i membri del gruppo riguardo ogni capitolato proposto. Queste riunioni saranno utili agli *Analisti* per carpire le conoscenze tecnologiche e le preferenze di ogni componente del team. È sempre compito di questi ultimi redigere uno *Studio di Fattibilità v3.0.0* dei capitolati basandosi su:

- **Dominio tecnologico e applicativo:** conoscenza delle tecnologie richieste e del dominio, esperienze precedenti con le problematiche poste dal capitolato;



- **Rapporto costi/benefici:** prodotti simili già sviluppati e presenti sul mercato, quantità di requisiti obbligatori, costo della realizzazione rapportato al risultato previsto;
- **Individuazione dei rischi:** comprensione dei punti critici della realizzazione, individuazione di eventuali lacune tecniche o di conoscenza del dominio dei membri del gruppo, analisi delle difficoltà nell'individuazione dei requisiti e loro verificabilità.
- **Scadenze Temporali:** breve simulazione del rispetto delle scadenze fissate dal progetto didattico, assumendo come variabili le conoscenze dei membri del gruppo e una stima dei principali requisiti.

Un'ultima riunione, a *Studio di Fattibilità v3.0.0* concluso, presenterà la decisione finale sul capitolato scelto. **N.B.:** ad ogni componente del gruppo non creerà alcun disagio il dover apprendere tecnologie differenti da quelle già acquisite in esperienze precedenti o grazie allo studio: in tal modo si andrà ad arricchire il bagaglio tecnico ed esperienziale di ogni componente del team.



## 3 Sviluppo

### 3.1 Analisi dei Requisiti

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

#### 3.1.1 Classificazione dei Requisiti

Compito degli *Analisti* sarà quello di stilare una lista di tutti quei requisiti che andranno ad emergere dal capitolato di progetto e da eventuali riunioni col proponente. Tali requisiti andranno catalogati per tipo ed importanza, secondo la seguente codifica:

R[importanza][tipo][codice]

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

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

Tale uso dei numeri renderà molto semplice l'ordinamento per importanza.

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

- F: Funzionale;
- Q: Di Qualità;
- P: Prestazionale;
- V: Vincolo.

- **Codice:** identifica la relazione gerarchica che c'è tra i requisiti di uno stesso tipo. C'è quindi una struttura gerarchica per ogni tipologia di requisito.

A partire da tutti i requisiti trovati, si andranno quindi a compilare le tabelle di tracciamento dei requisiti. Una volta completate le tabelle, sarà compito dei *Verificatori* accertarsi della consistenza del tracciamento effettuato.

#### 3.1.2 Casi d'Uso

Nell'Analisi dei *casi d'uso*<sub>G</sub>, denominati anche *use case* con l'acronimo UC, ogni UC dovrà presentare i seguenti campi:

- Codice identificativo;
- Titolo;
- Diagramma *UML*<sub>G</sub>2.x;
- Attori Primari;
- Attori Secondari;
- Scopo e Descrizione;
- Precondizione;
- Postcondizione;
- Flusso principale degli eventi;
- Scenari alternativi;



- Requisiti dedotti dal caso d'uso.

Come detto sopra, ogni UC sarà accompagnato da un codice univoco e identificativo, che segue il seguente formalismo:

UC[codice univoco del padre][codice progressivo del figlio]

Sarà compito degli *Analisti* identificare i *casi d'uso*<sub>G</sub>, procedendo sempre dal generale al particolare. Ogni *caso d'uso*<sub>G</sub> è accompagnato dalla sua forma grafica espressa in UML 2.x.

## 3.2 Progettazione

### 3.2.1 Specifica Tecnica

I *Progettisti* dovranno descrivere la progettazione ad alto livello dell'architettura dell'applicazione e dei singoli componenti nella *Specifica Tecnica v2.0.0*. Questo potrà essere possibile una volta accettati i requisiti accettati nel documento *Analisi dei Requisiti v3.0.0*. Il contenuto del documento in esame sarà specificato nei paragrafi successivi.

#### 3.2.1.1 Tecnologie Adottate

In tale paragrafo saranno descritte le tecnologie adottate nel progetto didattico; verranno redatti un elenco e una descrizione dei *framework*<sub>G</sub> che verranno utilizzati nella codifica dell'applicativo UMAP.

#### 3.2.1.2 Diagrammi UML

In tale sezione si andranno ad inserire i seguenti diagrammi:

- Diagrammi dei *package*<sub>G</sub>: sono molto utili per poter comprendere l'architettura generale; ad ognuno dei *package*<sub>G</sub> è associato un **codice univoco**. All'interno di un *package*<sub>G</sub> è possibile trovare classi o altri *package*<sub>G</sub> di grado gerarchico inferiore e vengono messe in evidenza le dipendenze (entranti e uscenti) tra di essi;
- Diagrammi delle classi: descrivono le entità (classi o interfacce) che compongono un *package*<sub>G</sub>. A differenza dei diagrammi sopra citati, è necessario indicare anche il tipo di dipendenza tra due entità (dipendenza semplice, associazione, aggregazione, composizione o ereditarietà). Si **raccomanda** di omettere i membri (alcuni attributi o metodi) della classe ritenuti dai *Progettisti* meno significativi rispetto all'architettura descritta, al fine che questa sia più chiara; tale raccomandazione vale anche per le dipendenze tra entità;
- Diagrammi di attività: aiutano a mostrare i vari flussi individuati in uno o più casi d'uso, esplicitando con un ordine logico e causale le azioni a cui essi fanno riferimento. In questi diagrammi vengono utilizzate primitive di alto livello quali *merge*, *branch*, *join*, i quali consentono di modellare azioni come scelte o elaborazione parallela;
- Diagrammi di sequenza: attraverso tali diagrammi è possibile illustrare il modo in cui le componenti di un sistema interagiscono tra loro, scambiandosi messaggi, per realizzare un certo scopo (e.g. accoppiamento tra client e server). Sebbene questo tipo di diagrammi disponga di primitive avanzate quali cicli, condizioni e frame (con diversi operatori, tra cui *alt*, *opt*, *region*, etc.), si raccomanda il *Progettista* di usare solamente le primitive basilari che offrono questi diagrammi, affinché la loro lettura sia più efficiente per tutti gli *stakeholders*<sub>G</sub>.

#### 3.2.1.3 Design Pattern

I *Progettisti* devono descrivere i *design pattern*<sub>G</sub> che andranno utilizzati nella realizzazione dell'architettura: di essi si dovrà andare a specificare:

- Nome del *pattern*<sub>G</sub> descritto;
- Tipo del *pattern*<sub>G</sub> (architetturale, comportamentale, strutturale o creazionale);

- Breve descrizione dello stesso;
- Contesto applicativo, facendo riferimento a componenti e/o entità appartenenti all'architettura del prodotto (oppure al framework scelto nel caso in cui il pattern sia di tipo architetturale).

#### 3.2.1.4 Tracciamento Componenti

Ogni requisito deve essere tracciato al componente che lo soddisfa. In questo modo sarà possibile misurare il progresso nell'attività di progettazione e garantire che ogni requisito venga soddisfatto.

#### 3.2.1.5 Test di Integrazione

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

### 3.2.2 Definizione di Prodotto

I *Progettisti* devono produrre la *Definizione di Prodotto v1.0.1* in cui viene descritta la progettazione di dettaglio del sistema, ampliando quanto scritto nella *Specifica Tecnica v2.0.0*. A tal fine, essi devono redigere una o più versioni del documento in esame, nel quale viene descritta l'architettura di dettaglio dei sottosistemi individuati nell'attività di progettazione architetturale. Tale documento sarà composto certamente delle seguenti sezioni:

- Introduzione;
- Standard di progetto adottati;
- Tracciamento delle componenti definite con la progettazione di dettaglio.

Una breve descrizione dei precedenti punti viene riportata nei successivi paragrafi.

#### 3.2.2.1 Introduzione

In questa prima sezione vengono esposti alcuni suggerimenti e consigli per la lettura del documento; in particolare, verranno specificati i riferimenti utilizzati per lo svolgimento dell'attività di progettazione di dettaglio e dovranno essere riportate le norme per distinguere le parole di glossario.

#### 3.2.2.2 Standard di Progetto

Nella sezione in esame, vengono dichiarati gli standard adottati nell'attività di progettazione di dettaglio. Sarà necessario sottolineare almeno:

- Standard di progettazione architetture adottati;
- Standard di documentazione del codice;
- Standard di denominazione di entità e relazione tra di esse;
- Standard di codifica;
- Strumenti di lavoro per la realizzazione del prodotto.

Sarà sempre possibile rimandare ad una versione delle *Norme di Progetto* in cui gli standard utilizzati vengono descritti con maggiore chiarezza.

#### 3.2.2.3 Definizione di Classe

Ogni classe prevista e progettata deve essere descritta nella *Definizione di Prodotto v1.0.1*; tale descrizione comprenderà l'elenco dei metodi e degli attributi, una breve spiegazione sullo scopo della classe e deve specificare le funzionalità che modella.



#### 3.2.2.4 Tracciamento delle Classi

Ogni requisito deve essere tracciato alle classi che lo soddisfano. In questo modo sarà possibile misurare il progresso nell'attività di progettazione e garantire che ogni classe soddisfi almeno un requisito.

#### 3.2.2.5 Test di Unità

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

### 3.3 Codifica

Tutti i file contenenti codice o documentazione dovranno essere conformi alla codifica *UTF-8<sub>G</sub>*.

#### 3.3.1 Nomi

I nomi di variabili, metodi, classi e funzioni dovranno essere scritti in Inglese. Il contenuto dei commenti sarà scritto invece in Italiano. I nomi di variabili, metodi e funzioni dovranno avere la prima lettera minuscola, i nomi delle classi dovranno avere la prima lettera maiuscola. Nel caso in cui ci siano nomi composti da più parole, non devono essere inseriti caratteri di separazione: infatti, ogni parola, dalla seconda in poi, sarà distinta dalla lettera maiuscola. Ad esempio possiamo avere il nome di un metodo in questa forma:

thisIsAMethod()

oppure il nome di una classe in questa forma:

ThisIsAClass{ ... }

#### 3.3.2 Ricorsione

La ricorsione va evitata nella maniera più assoluta, onde evitare un elevato consumo di memoria a discapito quindi delle performance del prodotto software.

#### 3.3.3 Documentazione del Codice

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

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

##### 3.3.3.1 Classi

Prima di ogni classe dovrà essere presente un commento contenente:

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



### 3.3.3.2 Metodi

Prima di ogni metodo/funzione dovrà essere presente un commento contenente:

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

Nel caso in cui si sentisse la necessità di spiegare codice di difficile comprensione, si potranno inserire dei commenti dalla riga precedente al codice che si vuole appunto commentare.



## Parte II

# Processi di Supporto





## 4 Documentazione

Questa sezione descrive tutte le convenzioni scelte e messe in pratica da LaTeXBiscotti riguardo la compilazione, la verifica e l'approvazione dei documenti da produrre.

### 4.1 Template

Per rendere più facile la stesura dei vari documenti sono stati realizzati una serie di template  $\text{\LaTeX}$  contenenti tutte le impostazioni stilistiche, di layout, tutti i comandi definiti da utente e le impostazioni locali riguardanti ciascun file. I file di impostazioni globali sono reperibili nel *repository*<sub>G</sub> LaTeXBiscotti/modello, e sono titolati `global.tex` e `layout.tex`; i file di impostazioni locali sono denominati `local.tex` e sono presenti in ogni *repository*<sub>G</sub> dei vari documenti redatti.

### 4.2 Struttura del Documento

#### 4.2.1 Frontespizio

Ogni documento redatto nella realizzazione del progetto è caratterizzato da un frontespizio così definito:

- Logo del Gruppo;
- Titolo del documento;
- Nome del Gruppo - Nome del Progetto;
- Versione del documento;
- Nome e cognome del/i *Redattore*/i del documento;
- Nome e cognome del/i *Verificatore*/i del documento;
- Nome e cognome del/i *Responsabile*/i approvatore/i del documento;
- Destinazione d'uso del documento;
- Lista di distribuzione del documento;
- Breve descrizione del contenuto del documento.

#### 4.2.2 Diario delle Modifiche

La seconda pagina del documento, o eventualmente anche la terza, contiene il diario delle modifiche del documento. Ogni riga della tabella che si viene a formare contiene:

- **Versione:** numero di versione dopo la modifica;
- **Data:** data della modifica;
- **Persone coinvolte:** persone che sono state coinvolte alla specifica modifica;
- **Descrizione:** breve descrizione delle modifiche effettuate al documento.

Tale tabella così composta è ordinata per data in ordine decrescente: si ha così che la prima riga conterrà la modifica più recente e quindi la versione attuale del documento.

#### 4.2.3 Indici

In ogni documento è presente un indice delle sezioni che lo compongono; nell'eventualità che in esso ci siano tabelle o figure, verranno composti i relativi indici.

#### 4.2.4 Formattazione delle Pagine

L'intestazione di ogni pagina del documento, escluso il frontespizio, contiene:

- Logo del gruppo;
- Nome del gruppo;



- Nome del progetto;
- Sezione corrente del documento.

Il piè di pagina, invece, contiene:

- Nome e versione del documento;
- Pagina corrente nel formato X di Y, dove X è il numero di pagina in cui ci si trova, Y è il numero di pagine totali del documento.

### 4.3 Norme Tipografiche

In questa sezione si vogliono indicare le convenzioni tipografiche e ortografiche in modo tale da mantenere uno stile uniforme su tutti i documenti inerenti al progetto.

#### 4.3.1 Punteggiatura e Norme Grammaticali

- **Punteggiatura:** un carattere di punteggiatura non deve mai seguirne uno di spaziatura e dopo ogni carattere di punteggiatura, eccetto l'apostrofo, deve esserci un carattere di spaziatura;
- **Parentesi:** quanto racchiuso tra due parentesi non deve mai iniziare e/o finire con un carattere di spaziatura;
- **Lettere maiuscole:** le lettere maiuscole sono da inserire dove lo prevede la lingua italiana e:
  - All'inizio di ogni elemento di un elenco puntato;
  - Per indicare il nome del gruppo di progetto;
  - Per indicare il nome dei ruoli;
  - Per indicare il nome dei documenti;
  - Per indicare il nome delle fasi di lavoro;
  - Per rimarcare le parole Committente e Proponente;
  - Nelle parole principali dei titoli dei vari paragrafi.

#### 4.3.2 Stile di Testo

- **Grassetto:** il grassetto può essere utilizzato nei seguenti casi:
  - **Elenchi puntati:** per evidenziare il concetto di cui si va a parlare in tale punto della lista;
  - **Casi particolari:** in situazioni eccezionali, per andare ad indicare concetti o parole chiave.
- **Corsivo:** il corsivo deve essere usato nelle seguenti occasioni:
  - **Citazioni;**
  - **Abbreviazioni;**
  - **Nomi particolari:** ad esempio *Progettista*;
  - **Documenti.**
- **Typewriter:** tale font serve per formattare nomi di classi o file, indirizzi web e percorsi;
- **Maiuscolo:** è consentito scrivere una parola completamente in maiuscolo solamente in caso di acronimi o nelle macro  $\LaTeX$ ;
- $\LaTeX$ : ogni riferimento a  $\LaTeX$  va scritto usando il comando `\LaTeX`.



#### 4.3.3 Composizione del Testo

- **Elenchi puntati:** ogni punto di un elenco puntato deve terminare con un punto e virgola, se si tratta dell'ultimo con un punto; ogni item dell'elenco deve iniziare con una lettera maiuscola, a meno di casi particolari, come, ad esempio, il nome di un file. Nel caso in cui il testo dei punti sia nel formato {Oggetto}: {descrizione} e la descrizione contenga alla fine un ulteriore elenco puntato il tutto terminerà con il punto di tale sotto-elenco puntato;
- **Nota a piè di pagina:** tale elemento deve iniziare con una lettera maiuscola per la prima parola, ad eccezione del caso in cui si tratti di un url, e questa non deve essere mai preceduta da un carattere di spaziatura.

#### 4.3.4 Formati

- **Indirizzi web:** per gli indirizzi web deve essere utilizzato il comando  $\text{\LaTeX}\backslash\text{url}$ ;
- **Date:** le date presenti in tutti i documenti devono seguire lo standard  $ISO_{\alpha}$  8601:2004:

AAAA-MM-GG

dove:

- *AAAA*: rappresentazione anno con quattro cifre;
- *MM*: rappresentazione mese con due cifre;
- *GG*: rappresentazione giorno con due cifre.
- **Comandi utente:** sono stati creati nel file `modello/global.tex` tutti quei comandi utente che possono rendere più fluida e comoda la scrittura dei documenti; si rimanda alla consultazione di tale file per utilizzarli al meglio. Ogni modifica che un membro del gruppo intende fare a tale file deve essere prima autorizzata dall'*Amministratore di Progetto*;
- **Nomi propri:** l'utilizzo dei nomi propri dei membri del team deve seguire la forma "Nome Cognome", a meno di una lista ordinata alfabeticamente, in cui è preferibile un ordinamento "Cognome Nome".

#### 4.3.5 Sigle

Le sigle dei documenti e delle revisioni sono da utilizzare solamente all'interno di tabelle o diagrammi, in modo da non appesantirne la lettura. Sono previste le seguenti sigle:

- **AdR** = Analisi dei Requisiti;
- **GL** = Glossario;
- **NdP** = Norme di Progetto;
- **PdP** = Piano di Progetto;
- **PdQ** = Piano di Qualifica;
- **SdF** = Studio di Fattibilità;
- **ST** = Specifica Tecnica;
- **RA** = Revisione di Accettazione;
- **RP** = Revisione di Progettazione;
- **RQ** = Revisione di Qualifica;
- **RR** = Revisione dei Requisiti.



## 4.4 Componenti Grafiche

### 4.4.1 Tabelle

Per tutte le tabelle presenti in ogni documento, eccetto il diario delle modifiche, deve essere presente una didascalia che ne spieghi sinteticamente il contenuto; questa deve inoltre essere marcata con un numero identificativo incrementale per la tracciabilità della stessa all'interno del documento.

Altre convenzioni o implementazioni speciali per qualche tabella devono essere motivate e spiegate all'inizio del documento in questione.

### 4.4.2 Immagini

Le immagini che si andranno ad inserire all'interno dei documenti devono essere nel formato  $SVG_G$  per garantirne una migliore qualità in caso di ridimensionamento. Se non dovesse essere possibile questo tipo di formato, è da preferire il formato  $PNG_G$ .

## 4.5 Classificazione Documenti

### 4.5.1 Documenti informali

Un documento informale è da considerarsi tale dal momento della sua creazione fino all'approvazione da parte del *Responsabile di Progetto*. Prima di questo momento, il documento va considerato prettamente a uso interno del gruppo.

### 4.5.2 Documenti formali

Un documento viene inteso come formale una volta passata l'approvazione del *Responsabile di Progetto*. Tale documento sarà quindi pronto per essere presentato ai richiedenti. Per poter definire tale un documento, esso deve aver seguito il percorso di verifica e validazione descritto nel *Piano di Qualifica v3.0.0* e in §5.2.

## 4.6 Glossario

Il Glossario è un documento di utilità che va a spiegare, in modo conciso e comprensibile, tutte quelle parole presenti negli altri documenti e facenti parte del contesto applicativo che potrebbero essere soggette a fraintendimenti. Tale documento dovrà essere aggiornato di pari passo con la stesura dei documenti.



## 5 Gestione della Configurazione

### 5.1 Versionamento

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

$$vX.Y.Z$$

dove:

- *X*: indica il numero crescente di uscite formali del documento; è limitato superiormente dal numero di revisioni effettuate; tale indice parte da 0, valore che sta ad indicare che il documento non è ancora stato sottoposto ad alcuna revisione formale;
- *Y*: indica il numero di modifiche sostanziali del documento, partendo da 0; in particolare si va ad incrementare questo valore ogni qual volta il documento viene sottoposto alla verifica e la supera con successo;
- *Z*: indica il numero di modifiche minori fatte al documento; tale indice parte da 1. Ad ogni modifica effettuata al documento, che corrisponde ad un'aggiunta di una voce nel diario delle modifiche, l'*Amministratore* o il *Verificatore* devono aggiornare l'indice seguendo una numerazione progressiva. Non viene fissato un limite superiore per tale indice.

Per citare correttamente un documento in un altro basta rifarsi ai comandi utente definiti in `modello/global.tex`. La citazione del documento *Analisi dei Requisiti*, ad esempio, avrà il seguente formato:

$$\textit{Analisi dei Requisiti v3.0.0}$$

### 5.2 Ciclo di Vita

Ogni documento può trovarsi all'interno di tre stati diversi: questi stati costituiscono il suo ciclo di vita. In particolare abbiamo questi stati:

- **In lavorazione:** stato in cui si entra dal momento della creazione del documento; vi si rimane per tutto il tempo necessario alla sua stesura;
- **Da verificare:** una volta ultimato il documento, esso viene preso in consegna dai *Verificatori*, i quali hanno il compito di rilevare e correggere tutti gli errori e/o imprecisioni sintattici e semantici;
- **Approvato:** terminata la fase di verifica, il documento deve essere approvato dal *Responsabile di Progetto*. L'approvazione ne sancisce lo stato finale per la corrente versione.

Ovviamente, ogni documento può attraversare ogni fase più di una volta, partendo sempre dallo stato iniziale. Una volta che si andrà a completare nuovamente il ciclo, viene definita una nuova versione del documento in questione.



## 6 Verifica e Validazione

### 6.1 Metriche

In questa sezione delle norme di progetto vengono definite delle metriche utili a misurare l'efficienza e l'efficacia con cui il gruppo avanza nelle varie fasi di sviluppo del prodotto. Le metriche adottate saranno quattro, improntate soprattutto ad analizzare gli errori e l'agilità dei membri del gruppo nello svolgere le varie mansioni che verranno loro assegnate.

- **Errori frequenti:** Questa metrica indica quali e con che frequenza certi errori capitano durante la stesura dei documenti;
- **Livello di automazione:** Identifica un valore relativo alla quantità di tasks che il gruppo riesce ad eseguire in maniera automatizzata attraverso tools, software e strumenti in generale;
- **Numero di compilazioni fallite:** Indica il numero di volte che viene compilato il codice prima che la compilazione risulti priva di errori;
- **Ore di lavoro extra:** Indicano il numero di ore di lavoro che eccedono le ore pianificate per completare un'attività previste nel diagramma di Gantt.

Il verificatore può agire sugli errori riscontrati nei seguenti modi:

- **Correzione immediata:** è richiesto che il Verificatore proceda autonomamente alla correzione dell'errore;
- **Aggiunta alla checklist:** è richiesto che il Verificatore aggiunga l'errore riscontrato alla checklist appropriata.

### 6.2 Verifica dei processi

Al termine di ogni fase del progetto il Verificatore dovrà effettuare un controllo per ogni macro-attività al fine di ricavare degli indici per la valutazione del lavoro svolto secondo le metriche stabilite in precedenza. Conseguentemente verrà ricavato un valore che rappresenterà l'efficienza con cui il lavoro ha svolto tale *task*.

### 6.3 Verifica dei Documenti

#### 6.3.1 Walkthrough

Tale tecnica di verifica dei documenti prevede una lettura critica ed approfondita dell'intero documento in esame. Può essere effettuata dai *Verificatori* o da altri membri del gruppo di lavoro, previa deroga da parte del *Responsabile di Progetto*.

Con questa metodologia di verifica si andranno a produrre una o più *liste di controllo*, che saranno utilizzate per le successive analisi tramite *Inspection*. Da notare che, per procedere correttamente con la tecnica in esame, occorre molta attenzione nell'accertarsi che:

- Non siano presenti nel documento errori di forma o composizione;
- Il contenuto del documento sia valido e consistente.

#### 6.3.2 Inspection

La suddetta tecnica potrà essere utilizzata soltanto una volta compilata la *lista di controllo*, consultabile in appendice A di tale documento. Si tratta di una lettura mirata del prodotto in esame, cercando solamente errori in base a quanto scritto nella lista. Tale metodologia consente una verifica molto rapida ed efficace, a patto si sia redatta una efficace *lista di controllo*.

### 6.4 Verifica del tracciamento dei requisiti

Il tracciamento requisiti-fonti/fonti-requisiti viene gestito da un tool sviluppato internamente al gruppo, Mnemosyne, in modo da rendere questa pratica il più possibile automatizzata. Di conseguenza, al Verificatore viene semplicemente richiesto un controllo visivo del corretto funzionamento del *tool*.

## 6.5 Verifica dei diagrammi UML

I diagrammi  $UML_G$  vengono costruiti mediante il software **Astah**, il quale permette di mantenere un buon livello di coerenza con lo standard  $UML_G$ . Di conseguenza al Verificatore viene solo chiesto di verificare manualmente la correttezza dei diagrammi presentati.

## 6.6 Verifica della progettazione

Il Verificatore deve assicurarsi che i vari moduli del progetto siano realizzabili da un singolo Programmatore in modo da rendere il lavoro più agevole.

## 6.7 Verifica del codice

Il Verificatore dovrà sottoporre il codice ad analisi che verranno effettuate mediante plugin del programma *Jenkins* di seguito riportati:

- **sbt plugin**: questo plugin permette di costruire progetti in Scala tramite il tool sbt;
- **Jenkins Cobertura Plugin**: permette di calcolare la percentuale di codice sottoposta a test;
- **Jenkins TAP Plugin**: utility per test generali sul codice;
- **Jenkins Violations Plugin**: questo plugin verifica la presenza di particolari violazioni nel codice XML;
- **HTML Publisher Plugin**: plugin per la pubblicazione di report html;
- **Checkstyle Plugin**: genera trend report per checkstyle, un programma di analisi per codice statico;
- **Join plugin**: plugin utile per la generazione di progetti con dipendenze a diamante.

## 6.8 Validazione

La validazione, processo con il quale si intende verificare se i requisiti e le aspettative degli  $stakeholder_G$  siano state soddisfatte, verrà svolto dal gruppo in vista della Revisione di Accettazione.

### 6.8.1 Manuali Utente

Tra i requisiti di qualità, individuati in *Analisi dei Requisiti v3.0.0*, vi è anche quello che prevede la fornitura di uno o più Manuali Utente, utili alla comprensione e all'utilizzo della piattaforma. Per la validazione di questi, sarà sufficiente una normale attività di verifica seguita dall'approvazione del *Responsabile di Progetto*.

### 6.8.2 Codice

Per eseguire validazione sulle applicazioni, tutte le soglie individuate nel *Piano di Qualifica v3.0.0* dovranno essere rispettate. Oltre a ciò, è necessario eseguire i test di sistema presenti nella tabella in appendice *Piano di Qualifica v3.0.0*, per poi riportarne l'esito sovrascrivendo la colonna apposita della stessa tabella.



## Parte III

# Processi Organizzativi





## 7 Gestione di Progetto

La gestione di progetto, dalla nascita alla sua conclusione, viene affidata al *Responsabile di Progetto*. Con lo scopo di garantire un corretto sviluppo delle varie attività inerenti al progetto, dovrà utilizzare degli strumenti che gli consentano di:

- Gestire e controllare le risorse;
- Pianificare, controllare e coordinare le varie attività;
- Analizzare e gestire i rischi;
- Elaborare i dati.

### 7.1 Pianificazione delle Attività

Per pianificare al meglio le attività correlate al progetto, il *Responsabile di Progetto* deve realizzare un diagramma di *Gantt<sub>G</sub>*, utilizzando lo strumento integrato nel tool fornito da **Teamwork**. Tale diagramma viene automaticamente generato da Teamwork una volta che si vanno ad inserire le varie attività, sotto forma di elenchi di compiti. Per riportare i diagrammi suddetti nel documento *Piano di Progetto v3.0.0*, verrà utilizzato lo strumento **Gantt Project**.

### 7.2 Coordinazione e Controllo delle Attività

Per coordinare e controllare le attività, il *Responsabile di Progetto* deve informare dell'avvenuta creazione del diagramma di *Gantt<sub>G</sub>* tutti i membri del gruppo tramite post nel forum citato in §8.1.1, in modo che questi abbiano la possibilità di adempiere efficacemente ai propri compiti, inserendone lo stato di avanzamento. Così facendo il *Responsabile di Progetto* sarà sempre al corrente dello stato di avanzamento del progetto stesso.

### 7.3 Gestione e Controllo delle Risorse

Per gestire e controllare le risorse, il *Responsabile di Progetto* deve utilizzare Teamwork come indicato in §9.4. Tale norma gli consente, inoltre, di controllare lo stato di avanzamento di ogni processo.

### 7.4 Analisi e Gestione dei Rischi

Durante l'avanzamento del progetto, il *Responsabile di Progetto* deve controllare sempre che i rischi descritti nel *Piano di Progetto v3.0.0* o eventuali nuovi rischi non vadano a verificarsi. Nel caso ciò dovesse accadere, dovrà attuare le contromisure ivi descritte e riportare gli effettivi riscontri.

### 7.5 Elaborazione dei Dati

Il *Responsabile di Progetto* deve sfruttare il software Calc, o in alternativa Excel, come descritto in §9.3.4, per elaborare i dati raccolti durante lo sviluppo del progetto e riportarli nel *Piano di Progetto v3.0.0*.

### 7.6 Delega

Qualora il *Responsabile di Progetto* partecipasse alla redazione di un documento, potrà delegarne l'approvazione ad un *Verificatore*.

### 7.7 Assegnazione di Attività

Per assegnare le attività alle risorse disponibili, il *Responsabile di Progetto*, o in alternativa un responsabile da lui delegato, dovrà seguire le procedure descritte in §9.4.2.

### 7.8 Gestione dei Cambiamenti

In seguito ad un errore trovato da un *Verificatore* e segnalato tramite una notifica nel forum del gruppo, il *Responsabile di Progetto*, o un suo delegato, dovrà assegnare la correzione come descritto sempre in §9.4, nella sezione riguardante i bug. Al termine della correzione, sarà compito del *Responsabile di Progetto*, o di un suo delegato, approvarne o meno la modifica; in caso di non accettazione, provvederà al rifacimento della correzione.



## 7.9 Ruoli di Progetto

Per tutta la durata del progetto, i componenti del gruppo andranno ad assumere diversi ruoli, rappresentativi di figure aziendali specializzate, indispensabili per una buona organizzazione interna e per il buon esito del lavoro. Ogni membro dovrà, a rotazione, assumere *almeno una volta* ogni ruolo, come specificato nei vincoli di organigramma. Per evitare possibili conflitti causati dalla rotazione dei ruoli, le attività principali assegnabili a specifici ruoli sono pianificate nel *Piano di Progetto v3.0.0*. Ogni componente del gruppo è tenuto a svolgere le attività assegnategli e a rispettare il ruolo che ne consegue. Il *Responsabile di Progetto* ha il compito di fare rispettare i ruoli assegnati durante le attività, mentre il *Verificatore* deve individuare le possibili incongruenze tra i ruoli e le modifiche registrate nei diari delle modifiche. Segue quindi una breve descrizione dei vari ruoli e delle varie mansioni che dovranno adempiere.

### 7.9.1 Amministratore

L'*Amministratore* equipaggia, organizza e gestisce l'ambiente di lavoro e di produzione, si occupa di redigere le *Norme di Progetto*. Le responsabilità che andrà ad assumere sono:

- Attuare le scelte tecnologiche concordate con il *Responsabile di Progetto*;
- Controllare versioni e configurazioni del prodotto;
- Risolvere i problemi legati alla gestione dei processi.

### 7.9.2 Analista

L'*Analista* è responsabile dell'attività di analisi. Per poter giungere correttamente a tale scopo deve comprendere a fondo il dominio applicativo. Redige lo *Studio di Fattibilità* e l'*Analisi dei Requisiti*, ovvero una specifica di progetto che cerca di comprenderne i vincoli e i rischi tecnologici. Non si occupa di trovare una soluzione al problema posto dal capitolato di progetto, ma cerca di estrarre il maggior numero di requisiti.

### 7.9.3 Progettista

Il *Progettista* è responsabile dell'attività di progettazione, ha una profonda conoscenza delle tecnologie che si andranno ad utilizzare ed è aggiornato alle ultime novità proposte da queste. Ha il compito di trovare una soluzione dati i requisiti forniti dall'*Analista*, e, una volta compiuta la scelta, se ne assume la piena responsabilità. Redige la *Specifica Tecnica*, la *Definizione di Prodotto* e la parte programmatica del *Piano di Qualifica*.

### 7.9.4 Programmatore

Il *Programmatore* ha responsabilità circoscritte: si occupa dell'attività di codifica nel rispetto delle *Norme di Progetto* e di quanto stabilito dal *Progettista*, senza spazio per fantasie o idee personali.

### 7.9.5 Responsabile di Progetto

Il *Responsabile* ha sempre l'ultima voce in capitolo per quanto concerne le decisioni sul progetto e ne è il responsabile ultimo dei risultati. Approva l'emissione di tutti i documenti e redige il *Piano di Progetto*. Le principali responsabilità che si deve assumere sono le seguenti:

- Pianificazione e organizzazione dello sviluppo del progetto, stima tempi e costi, e assegnazione delle attività ai componenti del gruppo;
- Riportare lo stato del progetto ai committenti;
- Analizzare i rischi che possono incorrere, monitorarli e prendere provvedimenti a riguardo;
- Stabilire una *way of working*<sub>G</sub> per ogni componente del gruppo, ai fini di un'influenza positiva delle performance del gruppo.

### 7.9.6 Verificatore

Il *Verificatore* organizza ed attua le attività di verifica e controlla che le attività siano conformi alle norme. Va a controllare la struttura grammaticale e semantica di ogni documento, andando a correggere eventuali



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



## 8 Struttura Organizzativa

### 8.1 Collaborazione

#### 8.1.1 Comunicazioni

##### 8.1.1.1 Comunicazioni Interne

Per quanto riguarda le comunicazioni interne, sono state pensate più metodologie:

- **Forum:** tale forum si trova all'interno dello strumento di *project management*<sub>G</sub> utilizzato dal gruppo, ovvero **Teamwork**<sup>1</sup>. Per tutti i membri del team sarà possibile creare nuovi post e dividerli per categorie, in modo tale da strutturare nella maniera migliore possibile i vari argomenti trattati. Usando questo strumento, si avrà uno storico di tutti gli argomenti trattati dal gruppo;
- **Messaggistica Istantanea:** è stata creata una chat all'interno della piattaforma *Telegram*<sub>G</sub>. Con tale mezzo sarà possibile scambiarsi rapidamente dei messaggi urgenti. Se le informazioni trattate dovessero essere di rilievo per lo sviluppo del progetto, i membri sono obbligati a creare un post nel suddetto forum;
- **Videoconferenza:** sarà possibile sfruttare il sistema *Skype*<sub>G</sub>. Verrà quindi, una volta terminata la conversazione, redatto un verbale e inserito sempre come post al già citato forum.

##### 8.1.1.2 Composizione Post

In questo paragrafo viene descritta la struttura che deve avere un post da inserire nel forum del gruppo.

#### Titolo

Il titolo deve essere chiaro ed esaustivo, possibilmente stringato e non confondibile con altri preesistenti. Dovrà inoltre esplicitare in modo semplice quale sarà il contenuto del post stesso. Dovranno essere evitati in ogni modo i post riconducibili ad altri all'interno del forum, o post che non hanno significato all'interno della categoria in cui verranno inseriti. In caso di errore, il membro del gruppo sarà tenuto ad eliminare tempestivamente il messaggio, in modo tale da non creare confusione.

#### Contenuto

Il post dovrà contenere tutte le informazioni necessarie a rendere facilmente comprensibile l'argomento trattato a tutti i membri del gruppo. Sarà consigliato l'utilizzo di elenchi puntati o numerati per identificare i vari argomenti del post.

#### Allegati

Sarà possibile inserire degli allegati pertinenti al post, qualora ve ne sia la necessità. Essi possono essere, ad esempio, verbali di una comunicazione via *instant messaging*<sub>G</sub> o videoconferenza.

#### Categoria

Tutti i post che andranno inseriti nel forum dovranno appartenere ad una categoria ben definita. Se un componente del gruppo non riesce ad identificarne una per il proprio messaggio, dovrà rivolgersi all'*Amministratore del Progetto*. Sarà compito di quest'ultimo decidere a che categoria associare il messaggio, e, in caso non ne esistesse nessuna consona, spetterà sempre a lui crearne una di nuova.

**N.B.:** A nessun membro eccetto l'*Amministratore del Progetto* e il *Responsabile del Progetto* sarà consentito creare nuove categorie all'interno del forum.

---

<sup>1</sup><https://www.teamwork.com/>



## Notifiche

Se un messaggio del forum dovesse essere rivolto ad un particolare membro del gruppo, può essergli associata una notifica personale. Se si tratta di un messaggio di comune interesse, ad ogni membro del gruppo dovrà essere associata una notifica.

### 8.1.2 Riunioni di Gruppo

#### 8.1.2.1 Frequenza

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

#### 8.1.3 Convocazione Riunione

##### Riunioni Generali

Il *Responsabile di Progetto* avrà il compito di convocare le riunioni generali, ovvero quelle in cui vengono convocati tutti i membri del gruppo; avrà anche il compito di valutare l'anticipazione della successiva riunione, in caso lo ritenesse necessario.

Qualora ve ne sia la necessità, qualsiasi componente del gruppo potrà richiedere la convocazione di una riunione generale; tuttavia, tale richiesta dovrà essere rivolta preventivamente al *Responsabile di Progetto*, il quale deciderà se accogliere o meno la richiesta.

Ogni riunione generale dovrà essere convocata, con almeno tre giorni di anticipo, tramite un post nel forum del gruppo, inserito sotto la categoria “Riunioni Generali”. Tale post dovrà essere così formato:

- **Titolo:** Convocazione Riunione n. X, dove X indica il numero progressivo di riunioni generali effettuate;
- **Contenuto:**
  - Data;
  - Orario;
  - Tipo: ordinaria/straordinaria;
  - Ordine del Giorno: elenco numerato delle varie voci da esaminare.

Ogni membro del gruppo dovrà confermare, o meno, la propria presenza tramite commento al post entro 36 ore. In caso di risposta negativa, dovrà essere fornita una giustificazione. Nell'eventualità che un membro non risponda entro il tempo utile, sarà compito del *Responsabile di Progetto* contattare telefonicamente il membro o i membri del gruppo interessato/i. Una volta ricevuta la risposta da parte di tutti i membri del gruppo, sempre il *Responsabile di Progetto* potrà decidere se confermare, annullare o spostare la riunione, in modo tale da permettere la partecipazione a tutti i componenti del team. In caso decida di spostare la riunione, dovrà commentare nuovamente il post con i dati relativi allo spostamento della riunione. In caso ciò avvenisse, i membri del gruppo sono tenuti a dare una risposta entro 12 ore.

##### Riunioni tra Alcuni Membri

Potranno essere necessarie alcune riunioni che andranno a coinvolgere la non totalità dei membri del gruppo. Ad esempio, in fase di Progettazione, è possibile e auspicabile la collaborazione tra *Analista* e *Progettista*: in caso ne sentissero la necessità, potranno indire una riunione che coinvolgerà solamente loro. Per fare ciò, si dovrà scrivere un post nella categoria dedicata (Riunioni tra membri) con la stessa dicitura dei post redatti per le riunioni generali. Per evitare confusione, dovranno essere notificati del post solo i diretti interessati alla riunione. Tutto quello che andrà discusso in tali riunioni dovrà essere verbalizzato, per mettere al corrente di quanto discusso tutti gli altri componenti del gruppo. Il verbale dovrà essere inserito, anche come allegato, ad un post da inserire in categoria “Verbali riunioni membri”.



### 8.1.3.1 Verbale

#### Riunione Interna

Il verbale di riunione interna è un documento interno informale che consente di tracciare gli argomenti discussi durante la riunione. Sarà scritto dal segretario della riunione, ruolo che andrà scelto di volta in volta, a rotazione, tra i membri del gruppo. Una volta redatto, il verbale dovrà essere inserito come post nell'apposita categoria del forum.

#### Riunione Esterna

In caso di riunione con il committente od il proponente, il verbale è un documento ufficiale che può avere valore normativo e quindi deve essere redatto seguendo criteri specifici, specificati in §4.

### 8.1.4 *Repository<sub>G</sub>* e Strumenti di Condivisione File

In questa sezione viene descritta la struttura del *repository<sub>G</sub>* dei documenti. Non è ancora stata definita una struttura per quella del codice. Tale decisione sarà presa nella fase di *Progettazione* dello sviluppo del progetto.

#### 8.1.4.1 Struttura *Repository<sub>G</sub>* dei Documenti

La struttura del *repository<sub>G</sub>*, la cui cartella principale viene denominata LaTeXBiscotti, è così composta:

- LaTeXBiscotti/documenti/{NomeDelDocumento}/  
Ciascuna cartella descritta da questo percorso contiene i file che vengono utilizzati dal documento {NomeDelDocumento}. In particolare conterrà il file {NomeDelDocumento}.pdf e il file `diario_modifiche.tex`, contenente il diario delle modifiche del documento;
- LaTeXBiscotti/ufficiali/  
Contiene i documenti ufficiali approvati dal *Responsabile di Progetto*;
- LaTeXBiscotti/modello/  
Contiene i file comuni a due o più documenti.

È raccomandato che tutti i file e le cartelle non contengano spazi nel loro nome. Non devono mai esserci due file o cartelle il cui percorso differisca soltanto per maiuscole/minuscole. Non bisogna inoltre rinominare file o cartelle modificandone soltanto il *case<sub>G</sub>* di alcuni caratteri del nome.

#### 8.1.4.2 Condivisione dei File

Per la condivisione informale di file e per il lavoro collaborativo su documenti di supporto, si usa la piattaforma di condivisione documenti prevista da Teamwork. Per una spiegazione più specifica si rimanda a §9.1.2.



## 9 Strumenti

### 9.1 Coordinamento

#### 9.1.1 Software di Versionamento

Il software scelto per la gestione del versionamento, dopo accurata analisi di altri prodotti (come SVN o Mercurial), è *GitHub<sub>G</sub>*. I motivi principali della scelta sono:

- **Interfaccia Grafica:** a differenza di *Git<sub>G</sub>* che prevede un utilizzo unicamente da linea di comando, *GitHub<sub>G</sub>* fornisce un'interfaccia grafica molto ben fatta e comprensibile per tutti i membri del gruppo; inoltre è possibile consultare online tutti i file inseriti nel *repository<sub>G</sub>*, senza fare una copia locale, ed è permesso controllare le differenze tra la versione ed un'altra di un singolo file, in modo da capire le differenze ed eventualmente risolvere i conflitti che si possono venire a creare;
- **Flessibilità:** *GitHub<sub>G</sub>* è un *repository<sub>G</sub>* distribuito che prevede la possibilità di *commit<sub>G</sub>* e *revert<sub>G</sub>* locali;
- **Servizi aggiuntivi:** *GitHub<sub>G</sub>* provvede a fornire, ad esempio, servizi di controllo accessi e *bug tracking<sub>G</sub>*, molto utili per la gestione efficiente del progetto;
- **Esperienza del Gruppo:** *GitHub<sub>G</sub>* è già stato usato da alcuni componenti del gruppo LaTeXeBiscotti.

##### 9.1.1.1 Workflow

Per un corretto utilizzo dello strumento di versionamento ogni membro del gruppo deve seguire la procedura di seguito riportata al fine di non creare conflitti all'interno dei documenti condivisi:

- La prima operazione da eseguire è la clonazione del *repository<sub>G</sub>* sulla macchina dove si intende modificare o creare i documenti attraverso il comando `git clone <indirizzo repositoryG>` tramite linea di comando;
- In seguito, ogni volta che si vorrà andare a lavorare sui file presenti in locale sulla propria macchina è previsto che venga eseguita un'operazione di aggiornamento all'ultima versione dei file in modo che combacino con quelli presenti nel *repository<sub>G</sub>*. Questa operazione è svolta mediante il comando `git pull` nella linea di comando;
- Dopo aver modificato uno o più file in locale è buona norma aggiornare anche la *repository<sub>G</sub>*. Per fare ciò bisogna navigare nella linea di comando fino a posizionarsi nella cartella in cui è stata clonata il *repository<sub>G</sub>* ed eseguire nell'ordine i seguenti comandi: `git add *` per aggiungere tutti i file modificati all'index di github sulla macchina locale, `git commit -m <messaggio>` per spostare i file modificati nello *head<sub>G</sub>* della macchina e infine `git push` per inviare i file alla *repository<sub>G</sub>*.

#### 9.1.2 Condivisione dei File

Si è scelto di usare come strumento di condivisione dei file il tool fornito dal software per la pianificazione e gestione del gruppo, descritto meglio nel paragrafo successivo. C'è la possibilità, infatti, di andare a condividere in un'apposita sezione "Documenti" tutti quei file che non necessitano di *controllo di versione<sub>G</sub>*, in modo da essere facilmente reperibili da tutti i membri del gruppo al momento del bisogno. Si potranno condividere appunti personali, manuali o altro, per andare ad uniformare le conoscenze di tutti i componenti del team.

### 9.2 Software di Pianificazione e Gestione dei Compiti

Per pianificare e gestire tutte le attività e i compiti annessi al progetto, si è deciso di utilizzare lo strumento Teamwork, programma gratuito funzionale in tutto e per tutto allo svolgimento del progetto. Si è deciso di utilizzare tale strumento per le seguenti caratteristiche:

- **Gestione dei singoli compiti:** i compiti possono essere facilmente inseriti ed assegnati ad un singolo membro del gruppo; per ogni compito è possibile definire una priorità, data di inizio e fine, aggiungere una descrizione ed inserire come allegati eventuali file; inoltre, il membro del gruppo responsabile del compito può aggiungere una percentuale di completamento del suddetto compito e il numero di ore lavoro che vi ha dedicato. Ogni compito può essere contrassegnato come chiuso una volta completato;



- **Gestione lista di compiti:** viene così implementato il concetto di attività come insieme di compiti: ad ogni lista è possibile inserire un qualsiasi numero di compiti;
- **Gestione pietre miliari:** nella stessa modalità della creazione di un singolo compito, è possibile creare una *milestone<sub>G</sub>*;
- **Diagrammi Gantt:** a partire dai compiti, dalle liste di compiti e dalle *milestone<sub>G</sub>*, viene generato automaticamente un diagramma di *Gantt<sub>G</sub>*, utile per una visione di insieme di tutte le attività e per evitare conflitti tra l'una e l'altra;
- **Calendario:** viene fornito un calendario che fornisce una vista d'insieme su tutte le attività da svolgere nel corso del mese selezionato; si può inoltre fruire di filtri che fanno visualizzare solamente i compiti assegnati ai singoli individui o le *milestone<sub>G</sub>* previste dal gruppo;
- **Condivisione Documenti:** vedi spiegazione in §9.1.2;
- **Forum:** vedi §8.1.1.

## 9.3 Strumenti per i Documenti

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

Per la stesura dei documenti si è scelto di utilizzare il linguaggio di markup L<sup>A</sup>T<sub>E</sub>X. Il motivo principale che ha portato a questa scelta è la facilità di separazione tra contenuto e formattazione: con L<sup>A</sup>T<sub>E</sub>X è possibile definire l'aspetto delle pagine in un file template condiviso da tutti i documenti. Altre soluzioni come Microsoft Office, LibreOffice o Google Docs non avrebbero consentito questa separazione, duplicando il lavoro di formattazione del testo e non garantendo un risultato uniforme. Il grande numero di pacchetti esistenti consente di implementare funzionalità comuni in maniera semplice. L'estensibilità di L<sup>A</sup>T<sub>E</sub>X può essere sfruttata per creare funzioni e variabili globali che rendono la scrittura del contenuto più corretta da un punto di vista semantico. Un esempio è dato dal comando `\GroupName` che va a scrivere il nome del gruppo. Per la scrittura di documenti L<sup>A</sup>T<sub>E</sub>X, l'editor consigliato è **TeXstudio**<sup>2</sup>; per gli utenti Mac OS X è consigliato, invece, l'utilizzo di **TeXShop**. Si veda inoltre §4.1 per informazioni sulla facilitazione alla scrittura dei documenti.

### 9.3.2 Mnemosyne

Per la stesura dell'Analisi dei requisiti il gruppo LaTeXeBiscotti si è affidato al programma online **Mnemosyne**, creato da Giovanni Rodighiero, membro stesso del gruppo, appositamente per gestire la stesura dei casi d'uso e il tracciamento automatico tra fonti, requisiti e casi d'uso. Dopo aver effettuato l'accesso tramite email e password, un membro autorizzato del gruppo si trova davanti al menù principale, dal quale può selezionare varie opzioni: le principali sono fonti, casi d'uso e requisiti. In ciascuna delle sezioni sopra citate è possibile visualizzare tutte le fonti, tutti i casi d'uso e i requisiti inseriti da ogni membro registrato; inoltre è possibile aggiungerne di nuovi, seguendo il formato di complicazione di ciascuno degli elementi. L'applicazione presenta un ottimo livello di automatizzazione nel collegare questi tre elementi, provvedendo al tracciamento automatico tra loro e fornendo la possibilità di esportare il lavoro svolto in essa in documenti L<sup>A</sup>T<sub>E</sub>X per la stesura definitiva del file. Nella sezione casi d'uso è inoltre possibile aggiungere in automatico un grafico *UML<sub>G</sub>*, creato in precedenza con il software **Astah**. La sezione test, in fase di sviluppo, per ora presenta la possibilità di effettuare test di sistema. Infine, la sezione glossario permette di, dato un insieme di parole inserite in *Glossario v3.0.0*, trovare tutte le occorrenze dei citati termini in un file di testo L<sup>A</sup>T<sub>E</sub>X e di apporre loro il tag *<sub>G</sub>*.

### 9.3.3 Verifica

Per la verifica dei documenti si utilizza TeXstudio, in quanto integra i dizionari di OpenOffice e segnala potenziali errori ortografici presenti nel testo, anche durante la stesura del testo stesso.

---

<sup>2</sup><http://www.texstudio.org/>



### 9.3.4 Diagrammi UML

Per quanto concerne la modellazione dei diagrammi dei caso d'uso, diagrammi di sequenza e diagrammi di attività è stato scelto l'editor **Astah Professional**<sup>3</sup>. Dopo aver valutato numerose altre alternative, tale strumento è risultato un ottimo software in grado di coniugare il completo supporto a *UML<sub>G</sub>* 2.0; si rilevano come pregi per tutti i componenti del team l'interfaccia semplice e molto usabile.

### 9.3.5 Fogli di Calcolo

Viene sfruttato il prodotto Calc del pacchetto LibreOffice, o in alternativa Excel del pacchetto Office, per produrre:

- 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.

### 9.3.6 Software per le Presentazioni

Il software ufficiale di cui il gruppo di lavoro farà uso per la preparazione delle presentazioni sarà **Prezi**<sup>4</sup>

## 9.4 Protocollo per lo Sviluppo dell'Applicazione

Per procedere con uno sviluppo controllato dei documenti e del codice si è scelto di adottare il sistema di gestione di progetto fornito da Teamwork. Tale scelta è stata motivata in §9.2. Si andranno ora a descrivere passo passo le varie operazioni possibili e consentite all'interno di tale portale.

Per quanto riguarda, invece, la segnalazione e l'assegnazione di bug, ritenendo che Teamwork non fornisca gli strumenti necessari per tale scopo, il gruppo ha deciso di adottare lo strumento delle *issue<sub>G</sub>* fornito da *Github<sub>G</sub>*. Si descrive al termine di questa sezione la modalità di creazione di una *issue<sub>G</sub>* e della sua assegnazione.

### 9.4.1 Creare un Nuovo Progetto

La creazione di un nuovo progetto è demandata al *Responsabile di Progetto*. Le operazioni da svolgere sono le seguenti:

- Selezionare l'opzione **Quick Add** nel menu superiore;
- Selezionare **Add a New Project**;
- Associare un **nome** significativo al progetto;
- Inserire una breve **descrizione** del progetto (facoltativo).

### 9.4.2 Assegnazione di un Nuovo Compito

Tale operazione è consentita al *Responsabile di Progetto* o ai *Verificatori*, in caso questi ultimi trovassero bug o errori durante la fase di verifica. Ogni compito deve essere inserito in una lista di compiti, la quale modula il concetto di attività, precedentemente creata. La creazione e l'assegnazione di un nuovo compito viene descritta nel diagramma di attività che segue.

---

<sup>3</sup><http://astah.net/>

<sup>4</sup><https://prezi.com/>

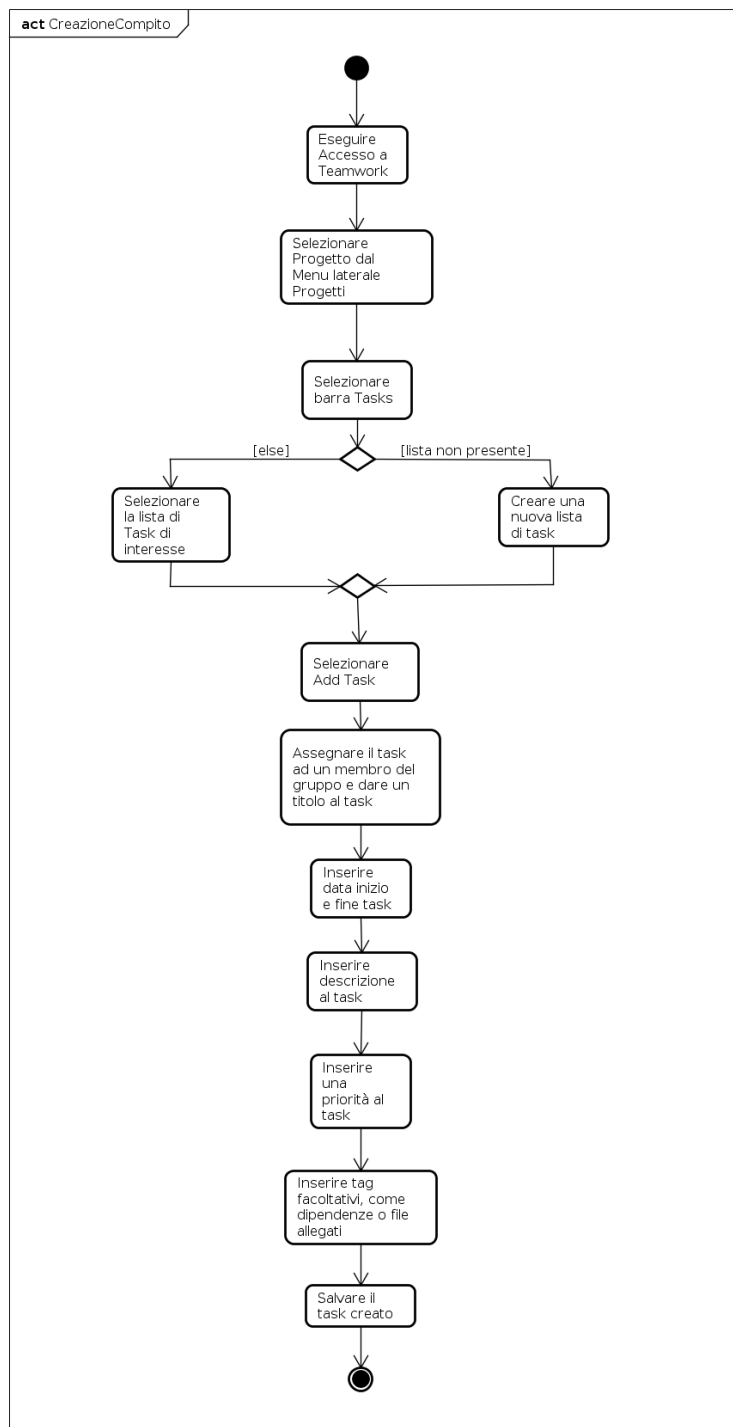


Figura 2: Diagramma attività - Creazione Nuovo Compito.

### 9.4.3 Modifica di un Compito

L'operazione di modifica è molto semplice ed intuitiva. Viene descritta dal diagramma di attività che segue.

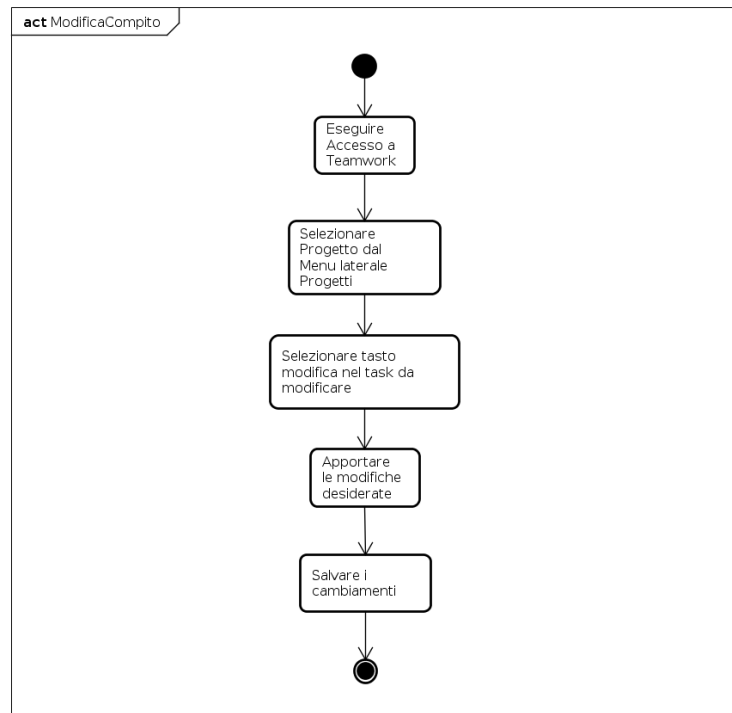


Figura 3: Diagramma attività - Modifica Compito.

#### 9.4.4 Segnalazione ed Assegnazione di una Issue (bug)

L'operazione sarà descritta tramite il diagramma di attività che segue. Tale operazione sarà permessa solo ai *Verificatori* e al *Responsabile di Progetto*. Gli stessi, dovranno creare la task su Teamwork, come precedentemente descritto. In tale diagramma si utilizza la terminologia legata al tool di *Github<sub>G</sub>* in riferimento al termine bug. Nell'operazione di inserimento di un commento significativo al bug rilevato, si andrà ad utilizzare la terminologia corretta riportata nel *Piano di Qualifica v3.0.0*.

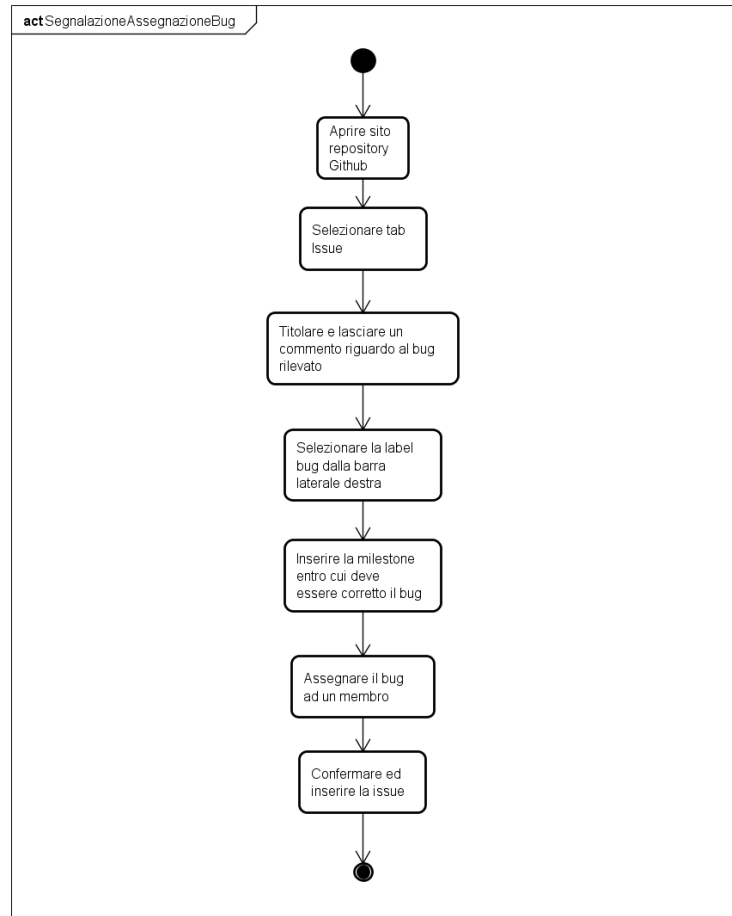


Figura 4: Diagramma attività - Segnalazione e Assegnazione Bug tramite Issue.

## 9.5 Strumenti per lo Sviluppo dell'Applicazione e per la Codifica

### 9.5.1 Framework

Per realizzare il progetto si fa uso del *framework<sub>G</sub> Play Framework<sub>G</sub>*<sup>5</sup>.

**IntelliJ IDEA**<sup>6</sup> è, invece, l'*IDE<sub>G</sub>* utilizzato per lo sviluppo. La scelta è stata operata per una serie di motivi, tra cui l'ottima integrazione con lo strumento di versionamento, la pulita e semplice interfaccia grafica e gli strumenti per il *debugging<sub>G</sub>*.

<sup>5</sup><https://www.playframework.com/>

<sup>6</sup><https://www.jetbrains.com/idea/>