

# MashUp

## Software Engineering Group

[info@mashup-unipd.it](mailto:info@mashup-unipd.it)

### Informazioni Documento

<b>Nome documento</b>	Norme di Progetto
<b>Versione</b>	v0.0.1
<b>Data redazione</b>	2014/12/10
<b>Redattori</b>	Santacatterina Luca Tesser Paolo
<b>Verificatori</b>	Roetta Marco
<b>Approvazione</b>	Cusinato Giacomo
<b>Lista distribuzione</b>	<i>MashUp</i>
<b>Uso</b>	Interno

### Sommario

Questo documento vuol definire le norme volte alla regolamentazione delle attività del gruppo *MashUp* necessarie al processo di sviluppo del progetto BDSMApp.

# Diario Revisioni

Modifica	Autore & Ruolo	Data	Versione
<i>Completamento capitolo Processi Organizzativi</i>	Tesser Paolo <i>Amministratore</i>	2014-12-26	v0.0.5
<i>Inizio stesura capitolo Processi Organizzativi</i>	Tesser Paolo <i>Amministratore</i>	2014-12-15	v0.0.4
<i>Inizio stesura capitolo Processi Primari</i>	Tesser Paolo <i>Amministratore</i>	2014-12-12	v0.0.3
<i>Inizio stesura capitolo Introduzione</i>	Santacatterina Luca <i>Amministratore</i>	2014-12-11	v0.0.2
<i>Inizio stesura del documento</i>	Tesser Paolo <i>Amministratore</i>	2014-12-10	v0.0.1

## Indice

<b>1 Introduzione</b>	<b>1</b>
1.1 Scopo del documento . . . . .	1
1.2 Scopo del prodotto . . . . .	1
1.3 Glossario . . . . .	1
1.4 Riferimenti . . . . .	1
1.4.1 Informativi . . . . .	1
1.4.2 Normativi . . . . .	2
<b>2 Processi Primari</b>	<b>3</b>
2.1 Processo di Fornitura . . . . .	3
2.1.1 Attività . . . . .	3
2.1.1.1 Accettazione . . . . .	3
2.1.1.1.1 Discussione e scelta del capitolo . . . . .	3
2.1.1.1.2 Studio di Fattibilità . . . . .	3
2.1.1.2 Preparazione della risposta . . . . .	3
2.1.1.2.1 Definizione e preparazione della proposta . . . . .	3
2.1.1.3 Pianificazione . . . . .	4
2.1.1.3.1 Scelta del modello di ciclo di vita . . . . .	4
2.1.1.3.2 Sviluppo e documentazione del Piano di Progetto . . . . .	4
2.2 Processo di Sviluppo . . . . .	4
2.2.1 Attività . . . . .	4
2.2.1.1 Analisi dei requisiti . . . . .	4
2.2.2 Norme . . . . .	4
2.2.2.1 Classificazione requisiti . . . . .	4
2.2.2.2 Classificazione casi d'uso . . . . .	4
2.2.3 Strumenti . . . . .	5
2.2.3.1 requirementsTool . . . . .	5
<b>3 Processi di Supporto</b>	<b>6</b>
3.1 Processo di documentazione . . . . .	6
3.1.1 Pianificazione . . . . .	6
3.1.1.1 Classificazione dei documenti . . . . .	6
3.1.1.1.1 Documenti informali . . . . .	6
3.1.1.1.2 Documenti formali . . . . .	6
3.1.1.2 Studio di fattibilità . . . . .	6
3.1.1.3 Norme di progetto . . . . .	6
3.1.1.4 Piano di progetto . . . . .	6
3.1.1.5 Piano di Qualifica . . . . .	6
3.1.1.6 Analisi dei requisiti . . . . .	6
3.1.1.7 Specifica tecnica . . . . .	6
3.1.1.8 Definizione di prodotto . . . . .	6
3.1.1.9 Glossario . . . . .	6
3.1.1.10 Verbali . . . . .	6
3.1.1.11 Manuale utente . . . . .	6
3.1.1.12 Procedure di gestione dei documenti . . . . .	6
3.1.1.12.1 Creazione di un documento . . . . .	6
3.1.1.12.2 Avanzamento di un documento . . . . .	6

3.1.1.13	Procedure di gestione del Glossario . . . . .	6
3.1.1.13.1	Inserimento di un termine . . . . .	6
3.1.1.13.2	Eliminazione di un termine . . . . .	6
3.1.2	Progettazione e sviluppo . . . . .	6
3.1.2.1	Versionamento . . . . .	6
3.1.2.2	Template . . . . .	6
3.1.2.3	Struttura dei documenti . . . . .	7
3.1.2.3.1	Prima pagina . . . . .	7
3.1.2.3.2	Diario delle modifiche . . . . .	7
3.1.2.3.3	Indici . . . . .	7
3.1.2.3.4	Formattazione di una pagina . . . . .	7
3.1.2.4	Suddivisione sezioni documenti . . . . .	7
3.1.2.4.1	Studio di fattibilità . . . . .	7
3.1.2.4.2	Norme di progetto . . . . .	7
3.1.2.4.3	Piano di progetto . . . . .	7
3.1.2.4.4	Piano di qualifica . . . . .	7
3.1.2.4.5	Analisi dei requisiti . . . . .	7
3.1.2.4.6	Specifica tecnica . . . . .	7
3.1.2.4.7	Definizione di prodotto . . . . .	7
3.1.2.4.8	Glossario . . . . .	7
3.1.2.4.9	Verbali . . . . .	7
3.1.2.5	Norme tipografiche . . . . .	7
3.1.2.5.1	Stile di testo . . . . .	7
3.1.2.5.2	Punteggiatura . . . . .	7
3.1.2.5.3	Composizione del testo . . . . .	7
3.1.2.5.4	Formati ricorrenti . . . . .	7
3.1.2.6	Componenti grafiche . . . . .	7
3.1.2.6.1	Immagini . . . . .	7
3.1.2.6.2	Tabelle . . . . .	7
3.1.3	Produzione . . . . .	8
3.1.3.1	Automatizzazione dei paragrafi . . . . .	8
3.2	Processo di verifica . . . . .	8
3.2.1	Pianificazione . . . . .	8
3.2.1.1	Tecniche di analisi . . . . .	8
3.2.1.1.1	Analisi statica . . . . .	8
3.2.1.1.2	Analisi dinamica . . . . .	8
3.2.2	Verifica . . . . .	8
3.2.2.1	Analisi statica del codice . . . . .	8
3.2.2.2	Analisi dinamica del codice . . . . .	8
3.2.2.3	Test di unità . . . . .	8
3.2.2.4	Verifica dei diagrammi . . . . .	8
3.2.2.5	Verifica della documentazione . . . . .	8
3.2.2.6	Comunicazione e risoluzione di anomalie e discrepanze	8
3.3	Processo di garanzia della qualità . . . . .	8
3.3.1	Qualità del prodotto . . . . .	8
3.3.2	Qualità del processo . . . . .	8

<b>4 Processi Organizzativi</b>	<b>9</b>
4.1 Gestione dei Processi . . . . .	9
4.1.1 Attività . . . . .	9
4.1.1.1 Gestione delle Comunicazioni . . . . .	9
4.1.1.1.1 Mail . . . . .	9
4.1.1.1.2 Comunicazioni interne . . . . .	9
4.1.1.1.3 Comunicazioni esterne . . . . .	9
4.1.1.2 Gestione delle Riunioni . . . . .	9
4.1.1.2.1 Riunioni interne . . . . .	9
4.1.1.2.2 Riunioni esterne . . . . .	10
4.1.1.3 Gestione dei task . . . . .	10
4.1.1.4 Gestione dello svolgimento dei task . . . . .	10
4.1.1.5 Gestione delle change request . . . . .	10
4.1.1.6 Gestione della commit . . . . .	10
4.1.1.6.1 Meccanismo di aggancio commit-task . . . . .	10
4.1.2 Procedure . . . . .	11
4.1.2.1 Procedura d'assegnazione dei task . . . . .	11
4.1.2.2 Procedura per il reperimento dell'id del task . . . . .	11
4.1.2.3 Procedura di svolgimento dei task assegnati . . . . .	11
4.1.2.4 Procedura di svolgimento di una change request . . . . .	13
4.1.2.5 Procedura di rilevazione dei rischi . . . . .	15
4.1.3 Norme . . . . .	15
4.1.3.1 Ruoli di Progetto . . . . .	15
4.1.3.1.1 Responsabile di Progetto . . . . .	17
4.1.3.1.2 Amministratore . . . . .	17
4.1.3.1.3 Analista . . . . .	17
4.1.3.1.4 Progettista . . . . .	17
4.1.3.1.5 Programmatore . . . . .	18
4.1.3.1.6 Verificatore . . . . .	18
4.1.3.2 Formato dei task . . . . .	18
4.1.3.3 Formato delle change request . . . . .	19
4.1.4 Strumenti . . . . .	19
4.1.4.1 Asana . . . . .	19
4.1.4.2 Astah . . . . .	20
4.1.4.3 ProjectLibre . . . . .	20
4.1.4.4 NetSons . . . . .	20
4.1.4.5 Skype . . . . .	20
4.1.4.6 WhatsApp . . . . .	20
4.1.4.7 TO DO - Strumento per la Presentazione . . . . .	20
4.2 Gestione delle Infrastrutture . . . . .	21
4.2.1 Attività . . . . .	21
4.2.1.1 Gestione del Repository . . . . .	21
4.2.1.2 Gestione dei Git Hooks . . . . .	21
4.2.1.3 Gestione del template del messaggio di commit . . . . .	21
4.2.2 Procedure . . . . .	21
4.2.2.1 Installazione e manutenzione Git Hooks . . . . .	21
4.2.2.2 Installazione e manutenzione messaggio di commit . . . . .	22
4.2.3 Norme . . . . .	23
4.2.3.1 Repository . . . . .	23
4.2.3.1.1 Nomi dei file in doc_BDSM_App . . . . .	23

4.2.3.1.2	Struttura di doc_BDSM_App . . . . .	23
4.2.3.1.3	Nomi dei file in src_BDSM_App . . . . .	25
4.2.3.1.4	Struttura di src_BDSM_App . . . . .	25
4.2.3.1.5	Modello di sviluppo . . . . .	25
4.2.3.1.6	Template messaggio di commit . . . . .	25
4.2.4	Strumenti . . . . .	26
4.2.4.1	Git . . . . .	26
4.2.4.2	GitHub . . . . .	26
4.2.4.3	Git Hooks . . . . .	26
4.2.4.4	Google Drive . . . . .	26
4.2.4.5	Sistema Operativo . . . . .	26
4.3	Formazione dei membri del team . . . . .	26

## 1 Introduzione

### 1.1 Scopo del documento

Questo documento vuole definire tutte le norme che i membri del gruppo *MashUp* dovranno rispettare durante lo svolgimento del progetto BDSMApp.

Tutti i membri sono tenuti a leggere il documento e a seguire rigorosamente le norme descritte.

Ogni membro del gruppo dovrà segnalare all'*Amministratore di Progetto* qualsiasi richiesta per la modifica o l'introduzione di nuove norme.

Successivamente l'*Amministratore di Progetto* coinvolgerà i membri del gruppo per discuterne e avrà a disposizione la possibilità di approvare o negare la richiesta. Tutte le norme raccolte dal seguente documento si riferiscono al progetto BDSMApp.

Le seguenti norme permetteranno di ottenere una maggiore uniformità dei documenti prodotti, porteranno ad un incremento dell'efficienza e della efficacia e ridurranno la possibilità di commettere errori.

In particolare si vuole definire:

- identificazione ruolo e mansione per ogni singolo componente del gruppo;
- strumenti e metodi per la comunicazione interna ed esterna al gruppo;
- strumenti e metodi per la stesura di tutti i documenti;
- tempi e modalità di lavoro per ogni singola fase;
- ambienti di sviluppo e tools di revisionamento.

### 1.2 Scopo del prodotto

Lo scopo del prodotto è di creare una nuova infrastruttura che permetta di interrogare Big Data recuperati dai social network, quali: Facebook, Twitter, Instagram. L'applicazione sarà composta da due parti:

- consultazione e interrogazione con interfaccia web per utente;
- servizi web REST interrogabili.

### 1.3 Glossario

Al fine di evitare ogni ambiguità relativa al linguaggio usato nei documenti viene allegato il *Glossario v0.0.1*.

Esso ha lo scopo di definire ed analizzare tutti i termini tecnici, di progetto e di ambiguità fornendo una ben precisa descrizione.

Tutte le occorrenze presenti nei documenti verranno contrassegnate con una “G” in formato pedice.

### 1.4 Riferimenti

#### 1.4.1 Informativi

- Asana <http://asana.com>
- BDSMApp <http://bit.ly/BDSMapp>
- GitHub <http://github.com>

#### 1.4.2 Normativi

- ISO 8601 [http://it.wikipedia.org/wiki/ISO\\_8601](http://it.wikipedia.org/wiki/ISO_8601)

## 2 Processi Primari

Nella sezione dei Processi Primari sono descritti i processi e le attività che servono alle parti prime durante il ciclo di vita del software.

Queste parti sono l'acquirente, il fornitore, lo sviluppatore, l'operatore e il manutentore del prodotto software.

Il gruppo *MashUp* si occuperà solo dei Processi di Fornitura e di Sviluppo. Il Processo di Acquisizione spetterà al proponente e al committente del capitolato scelto, mentre il Processo di Manutenzione non potrà essere attuato per vincoli temporali. Nonostante questo l'applicativo BDSMApp e la documentazione fornita con esso dovranno garantire la possibilità futura di procedere con le attività presenti in questo processo a discrezione del proponente.

### 2.1 Processo di Fornitura

#### 2.1.1 Attività

##### 2.1.1.1 Accettazione

**2.1.1.1.1 Discussione e scelta del capitolato** Il *Responsabile di Progetto* ha il compito di organizzare gli incontri iniziali per permettere ai componenti del gruppo di discutere e esaminare i capitolati selezionati dal committente.

Dalle riunioni emergerà la scelta effettuata dal team.

Le valutazioni che hanno portato ha prendere questa decisione verranno documentate in dei verbali.

**2.1.1.1.2 Studio di Fattibilità** Agli *Analisti* spetta il compito di redigere lo *Studio di Fattibilità v0.0.1*, basandosi sui verbali citati in precedenza. Per realizzare il documento dovranno essere presi in considerazione i seguenti punti per ogni capitolato, con maggiore dettaglio per quello preso in carico.

- **dominio tecnologico e applicativo:** si valutano le conoscenze attuali del team riguardo le tecnologie richieste;
- **interesse strategico:** si valuta quanto interesse abbia il team ad apprendere gli strumenti che serviranno a realizzare il capitolato in questione;
- **rischi possibili:** si analizzano i possibili punti critici in merito anche alle valutazioni effettuate nei punti precedenti.

##### 2.1.1.2 Preparazione della risposta

**2.1.1.2.1 Definizione e preparazione della proposta** I membri del gruppo *MashUp* dovranno redigere i seguenti documenti:

- *Studio di Fattibilità v0.0.1*
- *Analisi dei Requisiti v0.0.1*
- *Piano di Progetto v0.0.1*
- *Piano di Qualifica v0.0.1*

- *Norme di Progetto v0.0.1*

In allegato a essi si dovrà consegnare una *lettera di presentazione*. Starà quindi al committente e al proponente valutare l'offerta ricevuta e decidere se accettare o meno la proposta fatta.

### 2.1.1.3 Pianificazione

**2.1.1.3.1 Scelta del modello di ciclo di vita** Il *Responsabile di Progetto* avrà il compito di scegliere, se non indicato dal proponente, il modello di ciclo di vita adatto per lo sviluppo del prodotto richiesto.

**2.1.1.3.2 Sviluppo e documentazione del Piano di Progetto** Il *Responsabile di Progetto* dovrà eseguire una serie di compiti per delineare i lavori che i membri del team dovranno andare ad eseguire, calcolandone i costi e le tempistiche. Il tutto verrà documentato nel *Piano di Progetto v0.0.1*.

## 2.2 Processo di Sviluppo

### 2.2.1 Attività

**2.2.1.1 Analisi dei requisiti** Dopo avere redatto lo *Studio di Fattibilità*, agli *Analisti* spetterà il compito di scrivere il documento *Analisi dei Requisiti v0.0.1*. L'obiettivo sarà quello di cercare e produrre dei requisiti a partire dalle informazioni a disposizione del team.

Le risorse che potranno essere utilizzate per questo scopo provengono dal capitolato d'appalto e dagli incontri con il proponente e con il committente.

### 2.2.2 Norme

**2.2.2.1 Classificazione requisiti** TO DO (da completare una volta deciso come classificare i requisiti)

**2.2.2.2 Classificazione casi d'uso** I casi d'uso devono essere divisi in ordine gerarchico secondo il seguente schema:

UC[Codice]

- **Codice:** è il codice gerarchico e univoco per identificare ogni caso d'uso.

Per ogni caso d'uso dovranno essere presenti anche le seguenti informazioni:

- **descrizione:** fornire una breve descrizione del caso d'uso che sia il meno ambigua possibile;
- **attori principali:** elenco degli attori principali coinvolti nel caso d'uso;
- **attori secondari:** elenco degli attori secondari coinvolti nel caso d'uso;
- **scenari principali:** descrizione dei possibili scenari principali;
- **scenari alternativi:** descrizione dei possibili scenari secondari;

- **pre-condizioni:** condizione di partenza sempre vera all'inizio del caso d'uso;
- **flusso degli eventi:** ordine di esecuzione dei casi d'uso figli;
- **inclusioni:** spiegazione di tutte le inclusioni se presenti;
- **estensioni:** spiegazione di tutte le estensioni se presenti;
- **generalizzazioni:** spiegazione di tutte le generalizzazioni, se presenti;;
- **post-condizioni:** condizione finale sempre vera alla fine dell'esecuzione del caso d'uso.

### 2.2.3 Strumenti

#### 2.2.3.1 requirementsTool TO DO (da completare una volta generato l'applicativo)

## 3 Processi di Supporto

### 3.1 Processo di documentazione

#### 3.1.1 Pianificazione

##### 3.1.1.1 Classificazione dei documenti

###### 3.1.1.1.1 Documenti informali

###### 3.1.1.1.2 Documenti formali

##### 3.1.1.2 Studio di fattibilità

##### 3.1.1.3 Norme di progetto

##### 3.1.1.4 Piano di progetto

##### 3.1.1.5 Piano di Qualifica

##### 3.1.1.6 Analisi dei requisiti

##### 3.1.1.7 Specifica tecnica

##### 3.1.1.8 Definizione di prodotto

##### 3.1.1.9 Glossario

##### 3.1.1.10 Verbali

##### 3.1.1.11 Manuale utente

##### 3.1.1.12 Procedure di gestione dei documenti

###### 3.1.1.12.1 Creazione di un documento

###### 3.1.1.12.2 Avanzamento di un documento

##### 3.1.1.13 Procedure di gestione del Glossario

###### 3.1.1.13.1 Inserimento di un termine

###### 3.1.1.13.2 Eliminazione di un termine

### 3.1.2 Progettazione e sviluppo

#### 3.1.2.1 Versionamento

#### 3.1.2.2 Template

### 3.1.2.3 Struttura dei documenti

#### 3.1.2.3.1 Prima pagina

#### 3.1.2.3.2 Diario delle modifiche

#### 3.1.2.3.3 Indici

#### 3.1.2.3.4 Formattazione di una pagina

### 3.1.2.4 Suddivisione sezioni documenti

#### 3.1.2.4.1 Studio di fattibilità

#### 3.1.2.4.2 Norme di progetto

#### 3.1.2.4.3 Piano di progetto

#### 3.1.2.4.4 Piano di qualifica

#### 3.1.2.4.5 Analisi dei requisiti

#### 3.1.2.4.6 Specifica tecnica

#### 3.1.2.4.7 Definizione di prodotto

#### 3.1.2.4.8 Glossario

#### 3.1.2.4.9 Verbali

### 3.1.2.5 Norme tipografiche

#### 3.1.2.5.1 Stile di testo

#### 3.1.2.5.2 Punteggiatura

#### 3.1.2.5.3 Composizione del testo

#### 3.1.2.5.4 Formati ricorrenti

### 3.1.2.6 Componenti grafiche

#### 3.1.2.6.1 Immagini

#### 3.1.2.6.2 Tabelle

**3.1.3 Produzione**

**3.1.3.1 Automatizzazione dei paragrafi**

**3.2 Processo di verifica**

**3.2.1 Pianificazione**

**3.2.1.1 Tecniche di analisi**

**3.2.1.1.1 Analisi statica**

**3.2.1.1.2 Analisi dinamica**

**3.2.2 Verifica**

**3.2.2.1 Analisi statica del codice**

**3.2.2.2 Analisi dinamica del codice**

**3.2.2.3 Test di unità**

**3.2.2.4 Verifica dei diagrammi**

**3.2.2.5 Verifica della documentazione**

**3.2.2.6 Comunicazione e risoluzione di anomalie e discrepanze**

**3.3 Processo di garanzia della qualità**

**3.3.1 Qualità del prodotto**

**3.3.2 Qualità del processo**

## 4 Processi Organizzativi

### 4.1 Gestione dei Processi

#### 4.1.1 Attività

##### 4.1.1.1 Gestione delle Comunicazioni

**4.1.1.1.1 Mail** Ogni membro del gruppo avrà una mail personale creata grazie all'acquisizione di un dominio web sul servizio di hosting NetSons.

Il formato dell'indirizzo dovrà essere del tipo esposto di seguito e servirà per registrarsi ad ogni servizio che il team andrà ad utilizzare:

cognome.nome@mashup-unipd.it

**4.1.1.1.2 Comunicazioni interne** Le comunicazione interne e prettamente informali verranno gestite tramite un gruppo su WhatsApp denominato MashUp. Quelle formali avverranno attraverso il sistema di ticketing Asana che consente una chat di gruppo al suo interno e che notifica a tutti gli altri membri tramite mail quando qualcuno scrive qualcosa in essa.

Le norme e le procedure relative a questo servizio verranno trattate in dettaglio nella sezione 4.1.1.3 e in quella 4.1.4.1.

Se si necessita di un'interazione vocale con gli altri membri, qualora non fossero presenti nello stesso luogo, verrà utilizzata l'applicazione Skype.

**4.1.1.1.3 Comunicazioni esterne** Le comunicazione esterne vengono effettuate dal *Responsabile di Progetto* in quanto rappresenta il gruppo *MashUp*.

Egli, attraverso la seguente mail, manterrà i contatti con il proponente e con il committente. In caso lo ritenga necessario, girerà tali messaggi agli altri membri del team.

info@mashup-unipd.it

È stato inoltre stabilito, insieme al proponente, che l'interazione con lui, qualora non potesse avvenire tramite un incontro esterno, specificato nella sezione 4.1.1.2.2, possa avvenire tramite videochiamata in Skype e condividendo parte dei documenti del progetto.

##### 4.1.1.2 Gestione delle Riunioni

**4.1.1.2.1 Riunioni interne** Il *Responsabile di Progetto* ha il compito di convocare le riunioni interne al team. Dovrà quindi informare i componenti tramite le metodologie viste nella sezione 4.1.1.1.2.

Per ogni nuovo incontro dovranno essere specificati la data, l'ora, il luogo, il proponente e la motivazione che lo hanno reso necessario.

Ad ogni membro del gruppo è consentito chiedere la convocazione di una riunione interna. Il *Responsabile di Progetto*, una volta valutati i motivi e la necessità di tale incontro, provvederà ad organizzarlo secondo le norme viste in precedenza.

**4.1.1.2.2 Riunioni esterne** Il *Responsabile di Progetto* ha il compito di concordare la data, l'ora e il luogo dell'incontro con il proponente o con il committente attraverso il meccanismo visto nella sezione 4.1.1.3.

Una volta trovato l'accordo dovrà notificarlo agli altri membri secondo i metodi presenti nella sezione 4.1.1.2.

Ad ogni membro del gruppo è consentito chiedere la convocazione di una riunione esterna. Il *Responsabile di Progetto*, oltre ad accertarsi dei motivi e delle necessità di tale incontro, dovrà garantire che siano presenti almeno due componenti del team. Sarà compito di uno dei presenti, delegato di volta in volta, redigere il verbale dell'incontro avvenuto.

**4.1.1.3 Gestione dei task** Il *Responsabile di Progetto* deve pianificare e assegnare i compiti che i singoli membri dovranno andare a eseguire. Per farlo viene usato l'applicazione web Asana che consente di creare facilmente i task.

Permette inoltre un buon monitoraggio dell'andamento globale del progetto, offrendo grafici che illustrano il rapporto tra i ticket completati e quelli ancora da terminare in relazione anche alle milestone fissate dal team.

**4.1.1.4 Gestione dello svolgimento dei task** Ogni membro del team è tenuto ad accettare tutti i task che il *Responsabile di Progetto* pianificherà di assegnarli.

Se per dei motivi eccezionali l'assegnatario non potesse svolgere il compito indicato dovrà renderlo noto nei modi previsti nella sezione 4.1.1.1.2 al *Responsabile di Progetto*, il quale, una volta valutate le motivazioni, dovrà provvedere a trovare un nuovo destinatario del task.

Qualora il *Responsabile di Progetto* non ricevesse alcuna notifica entro le prime 24 ore riterrà automaticamente accettato il task dall'assegnatario.

**4.1.1.5 Gestione delle change request** (NOTA: DA SPOSTARE NEL PROCESSO DI SUPPORTO -> Risoluzione di Problemi)

Ogni membro del team può proporre dei cambiamenti o degli avanzamenti da apportare. Il suggerimento dovrà essere motivato adeguatamente in modo che il *Responsabile di Progetto* non debba preoccuparsi di ricercare le fonti che forniscono l'effettiva validità di questo mutamento.

Una volta verificata o meno la necessità dell'idea, il *Responsabile di Progetto* approverà o respingerà la proposta fatta.

Qualora la proposta fosse accettata il *Responsabile di Progetto* avrà il compito di pianificare le attività che serviranno ad apportare il cambiamento.

**4.1.1.6 Gestione della commit** Per fornire al *Responsabile di Progetto* un più ampio quadro di come stanno procedendo i singoli compiti viene utilizzato un servizio web offerto da GitHub.

Questo permette di mostrare nei commenti dei task il messaggio inserito durante la commit una volta effettuata l'operazione di push sul server.

Fornisce quindi il tracciamento del lavoro che viene svolto per portare a compimento il task in questione. Il servizio viene attivato sia in doc\_BDSM\_App sia in src\_BDSM\_App.

**4.1.1.6.1 Meccanismo di aggancio commit-task** È stato creato un account generale su Asana con user name "MainMashUp" che fornisce l'API\_Key necessaria a configurare il servizio senza che esso dipenda dall'account di un membro

in particolare.

Per permettere il tracciamento bisogna usare l'identificato del task di cui vogliamo tracciare i messaggi di commit.

L'id si può reperire nel modalità illustrate dalla figura 2 e va inserito all'interno del messaggio nelle modalità indicate nella sezione 4.2.3.1.6

#### 4.1.2 Procedure

**4.1.2.1 Procedura d'assegnazione dei task** Il *Responsabile di Progetto* dovrà eseguire i seguenti passi, riassunti in seguito nella figura 1.

Per effettuare più velocemente un'assegnazione si potrà servire del template illustrato nella figura 5.

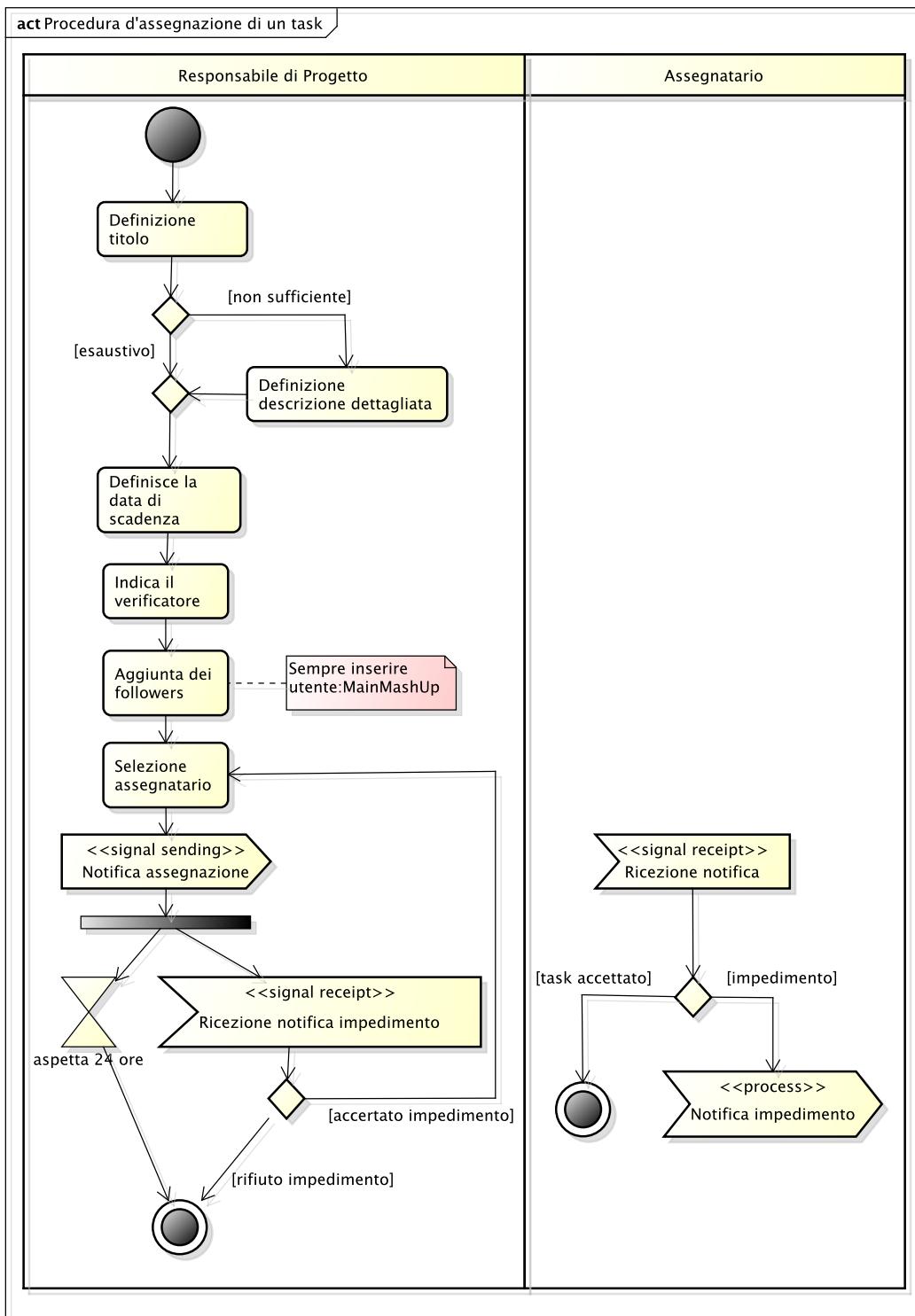
1. definire il titolo del task secondo le norme presenti nella sezione 4.1.3.2;
2. se necessario, fornire una descrizione più esaustiva del compito da svolgere;
3. definire la data di scadenza entro la quale il task dovrà essere portato a termine;
4. indicare il verificatore;
5. aggiungere tra i followers del task l'utente: “MainMashUp”;
6. selezionare l'assegnatario del task. Questa assegnazione genererà in automatico una notifica al membro scelto;
7. se si dovesse verificare il caso in cui l'assegnatario rifiutasse il task assegnato e il *Responsabile di Progetto* accertasse questa impossibilità nel suo svolgimento, dovrà eseguire di nuovo il passo precedente.

**4.1.2.2 Procedura per il reperimento dell'id del task** Ogni membro che vorrà recuperare l'identificativo del task su cui sta lavorando dovrà:

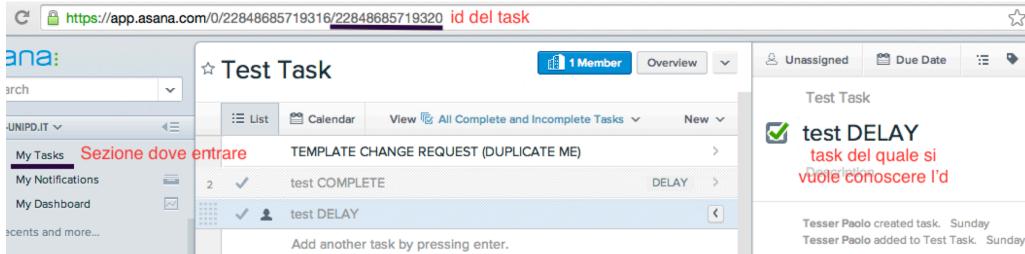
1. entrare su Asana;
2. posizionarsi sul task che si sta lavorando dalla sezione “MyTasks”;
3. posizionarsi sull'url e copiare tutte le cifre dopo l'ultimo slash “/”. Quello è l'id che si sta cercando.

**4.1.2.3 Procedura di svolgimento dei task assegnati** Il membro assegnatario del task, una volta ricevuta la notifica e non avendo nessuna motivazione per rifiutarlo, dovrà:

1. eseguire il task con scadenza più immediata;
  2. se dopo avere iniziato l'esecuzione del task, si riceve la notifica di uno nuovo con scadenza più immediata, si dovranno sospendere le attività che portano al compimento del task con scadenza meno ravvicinata ed eseguire nuovamente il passo precedente;
  3. se per svolgere il compito si dovesse andare oltre la data di scadenza prevista, si deve impostare il tag “DELAY” dal sistema Asana o tramite i modi espressi nella sezione 4.2.3.1.6.
- Questa situazione si può verificare se:



**Figura 1:** Diagramma di attività - procedura d'assegnazione di un task



**Figura 2:** Reperimento dell'id del task

- il tempo assegnato dal *Responsabile di Progetto* è inferiore a quanto richiesto per il compimento del task;
- il task può essere completato solo quando un altro task sia portato a termine. Queste dipendenze devono verificarsi il meno possibile ed è compito del *Responsabile di Progetto* evitare ciò in fase di pianificazione;
- l'assegnatario ha rallentamenti esterni non resi noti al *Responsabile di Progetto* in fase di pianificazione;
- il compimento del task necessita di competenze non in possesso dell'assegnatario.

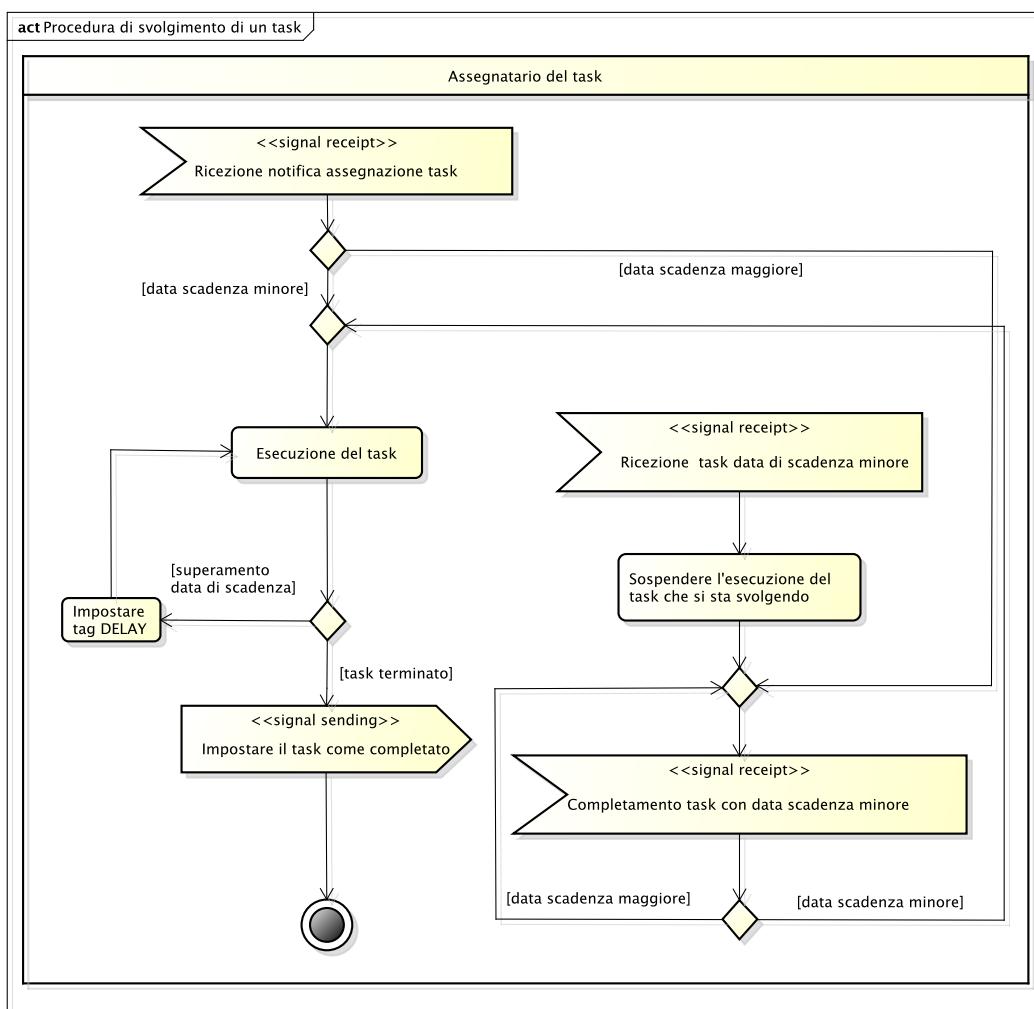
Nei primi due casi la responsabilità è imputabile al *Responsabile di Progetto*, mentre negli ultimi due la responsabilità è del membro che devo svolgere il task;

4. al completamento del task, dovrà chiudere il task, spuntandolo direttamente dal sistema Asana o tramite i modi espressi nella sezione 4.2.3.1.6;
5. l'operazione illustrata al passo precedente genera in automatico una notifica verso il *Verificatore* e il *Responsabile di Progetto*;
6. ritornare al primo punto con i task rimanenti.

#### 4.1.2.4 Procedura di svolgimento di una change request (NOTA: DA SPOSTARE NEL PROCESSO DI SUPPORTO -> Risoluzione di Problemi)

Ogni membro del gruppo, che intenda proporre dei cambiamenti o introdurre degli avanzamenti, è tenuto a seguire questa procedura .

Per effettuare più velocemente l'apertura di una change request si potrà servire del template illustrato nella figura 6.



**Figura 3:** Diagramma di attività - procedura di svolgimento dei task assegnati

1. definire il titolo del cambiamento/avanzamento;
2. definire una descrizione dettagliata del motivo che ha portato alla proposta e le fonti che garantiscono la bontà del cambiamento/avanzamento che si vuole attuare;
3. selezionare il *Responsabile di Progetto* in carica. Questa assegnazione genera in automatico una notifica.

Il *Responsabile di Progetto*, ricevuta la notifica, dovrà:

1. verificare le fonti e accertarsi della reale necessità di apportare il cambiamento o l'avanzamento proposto;
2. accettare o rifiutare la proposta;
3. notificare la decisione presa al proponente;
4. se la proposta è stata accettata, pianificare le attività che serviranno per apportarla.

**4.1.2.5 Procedura di rilevazione dei rischi** Il *Responsabile di Progetto* ha il compito di individuare i rischi trovati nel *Piano di Progetto v0.0.1*.

Qualora si trovasse davanti a problemi non previsti dovrà eseguire la seguente sequenza di attività:

1. registrazione del riscontro effettivo dei rischi nel *Piano di Progetto v0.0.1*;
2. pianificazione per la gestione dei nuovi rischi trovati;
3. aggiornamento delle metodologie per fare fronte alla nuova pianificazione attuata.

### 4.1.3 Norme

**4.1.3.1 Ruoli di Progetto** Lo sviluppo di un progetto necessita della collaborazione di diverse persone che andranno a ricoprire incarichi che rappresentano delle figure aziendali.

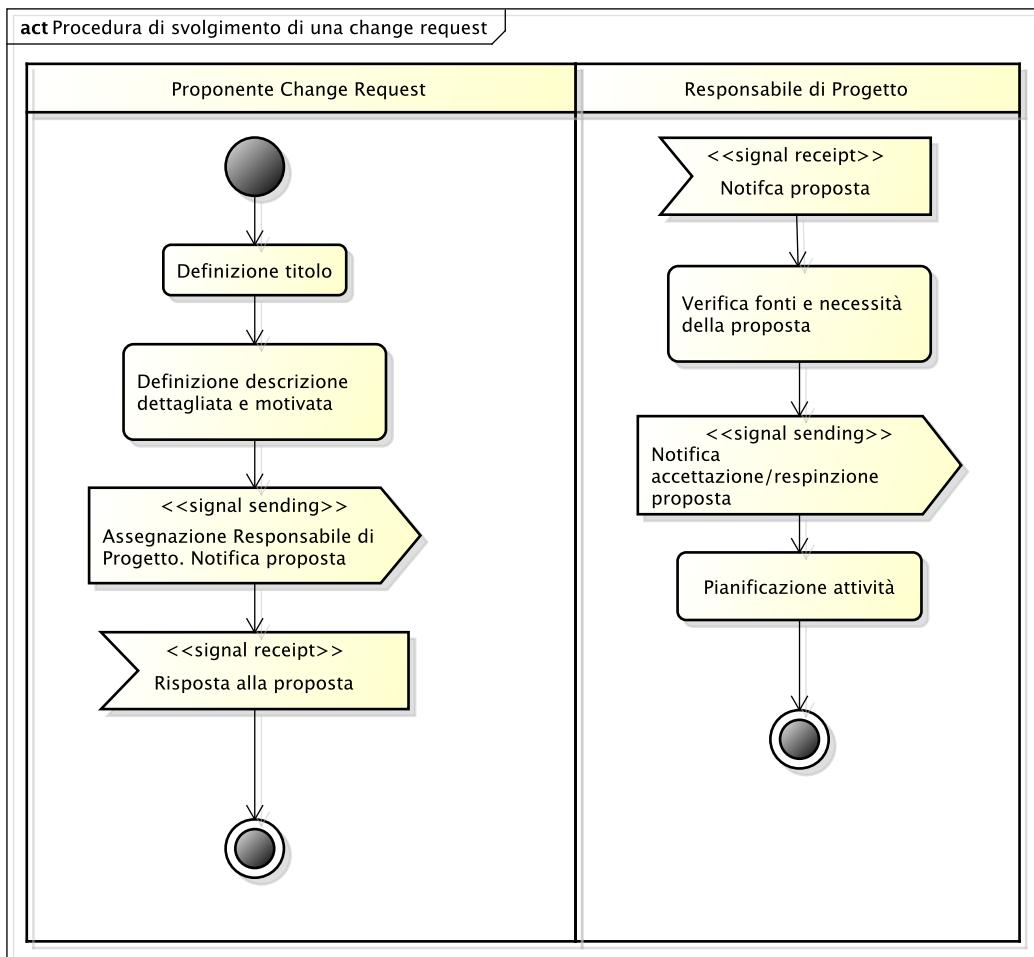
Viene garantito che ogni componente del gruppo *MashUp* dovrà ricoprire, almeno una volta, tutti i ruoli.

Questa rotazione potrebbe generare dei problemi dovuti a conflitti d'interesse in quanto lo stesso soggetto che svolge un compito potrebbe ritrovarsi a verificare l'operato svolto in precedenza.

Un'attenta pianificazione da parte del *Responsabile di Progetto* dovrà preoccuparsi di non fare avvenire queste situazioni.

Il compito di garantire che non siano stati fatti errori spetta al *Verificatore*, il quale, una volta riscontrata una incongruenza, dovrà notificarla al *Responsabile di Progetto* che avrà il compito di risolverla.

Vengono ora presentati i diversi ruoli, delineandone le mansioni e le responsabilità.



**Figura 4:** Diagramma di attività - procedura di svolgimento di una change request

**4.1.3.1.1 Responsabile di Progetto** Il *Responsabile di Progetto* rappresenta il team e il progetto verso il committente e il proponente.

Ha quindi le seguenti responsabilità e compiti:

- pianificazione e coordinamento delle attività;
- gestione e controllo delle risorse;
- analisi e gestione dei rischi;
- approvazione dell'offerta economica;
- assicurarsi che tutte le attività svolte siano conformi alle *Norme di Progetto v0.0.1* e rispettino la pianificazione effettuata nel *Piano di Progetto v0.0.1*;
- garantire che non ci siano conflitti di interesse. A tal proposito, se il *Responsabile di Progetto* dovesse prendere parte alla stesura di qualche documento, dovrà nominare un *Responsabile di Progetto* delegato che avrà il compito di sostituirlo nell'approvazione di quei documenti;
- redigere il *Piano di Progetto v0.0.1*.

**4.1.3.1.2 Amministratore** L'*Amministratore di Progetto* è il responsabile dell'ambiente di lavoro.

Ha quindi il compito di:

- ricercare e implementare strumenti che automatizzino il maggior numero di operazioni;
- gestire il versionamento del codice e della documentazione di progetto;
- fornire procedure che servono a garantire la qualità del prodotto uscente da un determinato compito;
- redigere le *Norme di Progetto v0.0.1*.

**4.1.3.1.3 Analista** L'*Analista* è il responsabile dell'attività di analisi.

Ha quindi il compito di:

- comprendere a pieno la natura del problema e la sua complessità;
- ricercare i requisiti che servono per realizzare il prodotto richiesto dal proponente;
- redigere lo *Studio di Fattibilità v0.0.1*;
- redigere l'*Analisi dei Requisiti v0.0.1*.

**4.1.3.1.4 Progettista** Il *Progettista* è il responsabile delle attività di progettazione.

Ha quindi il compito di:

- effettuare scelte progettuali estratte da soluzioni note e fortemente testate (design pattern);
- effettuare scelte progettuali che permettano al sistema di essere il più facilmente mantenibile in futuro;

- produrre una soluzione comprensibile, attuabile e motivata;
- redigere il *Piano di Qualifica v0.0.1*.

**4.1.3.1.5 Programmatore** Il *Programmatore* è il responsabile delle attività di codifica necessarie a portare il prodotto in uno stato che riesca a soddisfare i requisiti richiesti. Deve inoltre programmare i componenti che servono a verificare il sistema.

Ha quindi il compito di:

- scrive codice che rispetti le metriche stabilite per la sua scrittura;
- implementare lo soluzione descritte dal *Progettista*;
- implementare i test sul codice prodotto.

**4.1.3.1.6 Verificatore** Il *Verificatore* è il responsabile delle attività di verifica.

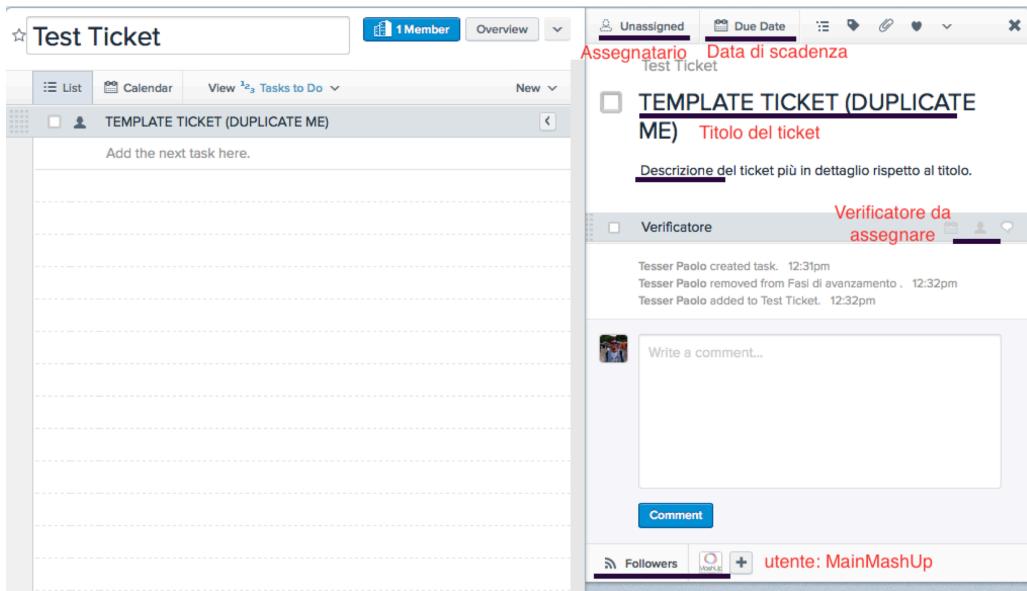
Ha quindi il compito di:

- controllare la conformità di ogni stadio del ciclo di vita del prodotto;
- garantire che le attività svolte siano conformi alle norme stabilite;
- redige il *Piano di Qualifica v0.0.1* per la parte che illustra l'esito e la completezza delle verifiche effettuate.

**4.1.3.2 Formato dei task** I task che si andranno ad aprire devono contenere le seguenti informazioni:

- **Titolo:** una breve descrizione del compito da eseguire;
- **Descrizione:** qualora non fosse abbastanza esaustivo il titolo del task, si deve fornire una descrizione più dettagliata dell'attività da svolgere;
- **Data di scadenza:** la data entro la quale dovrà essere svolto il lavoro;
- **Assegnatario:** il membro del team che dovrà eseguire il task indicato;
- **Verificatore:** il membro del team che andrà ad eseguire la verifica sull'operato svolto dall'assegnatario;
- **Followers:** i membri del team che si vuole rendere partecipi sull'attività svolta. Dovrà essere sempre aggiunto l'utente "MainMashUp".

La figura 5 illustra come vengono rappresentate queste informazioni su Asana. Inoltre l'immagine viene proprio presa da uno screenshot di un template di task, copiabile dal *Responsabile di Progetto* e modificabile una volta copiato a seconda dell'esigenze riportate in precedenza.



**Figura 5:** Template Task su Asana

#### 4.1.3.3 Formato delle change request (NOTA: DA SPOSTARE NEL PROCESSO DI SUPPORTO -> Risoluzione di Problemi)

Le change request che si andranno ad aprire devono contenere le seguenti informazioni:

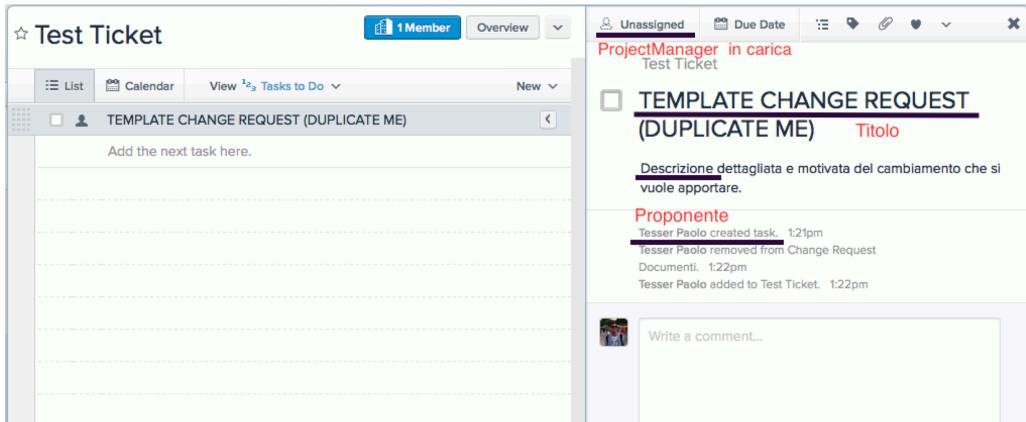
- **Titolo:** una breve descrizione di cosa si vorrebbe cambiare;
- **Descrizione:** una descrizione più dettagliata del cambiamento, motivando opportunamente la necessità riscontrata;
- **Assegnatario:** il *Responsabile di Progetto* al momento in carica secondo la rotazione dei ruoli;
- **Proponente:** chi è stato a proporre questo cambiamento. Sarà di fatto chi ha aperto la richiesta.

La figura 6 illustra come vengono rappresentate queste informazioni su Asana. Inoltre l'immagine viene proprio presa da uno screenshot di un template di una change request, copiabile da qualunque membro desideri proporre un cambiamento e modificabile una volta copiato a seconda dell'esigenze riportate in precedenza.

#### 4.1.4 Strumenti

##### 4.1.4.1 Asana

Asana è l'applicazione web scelta per la gestione dei task. Registrandosi con una mail di dominio personale come quella riportata nella sezione 4.1.1.1 è possibile usufruire di maggiori servizi che consentono una più facile gestione del team da parte del *Responsabile di Progetto*.



**Figura 6:** Template Change Request su Asana

**4.1.4.2 Astah** Astah è l'applicativo scelto per la creazione di grafici UML. La versione adottata è quella Professional, resa disponibile con una licenza gratuita per gli studenti dei corsi universitari registrandosi tramite l'indirizzo:

nome.cognome.X@studenti.unipd.it

**4.1.4.3 ProjectLibre** ProjectLibre è il prodotto scelto per la realizzazione dei diagrammi di Gaant e quelli di PERT.

**4.1.4.4 NetSons** NetSons è il servizio di hosting che il team ha deciso di adottare a scopo principalmente formativo.

Il dominio creato è:

<http://www.mashup-unipd.it>

Il servizio fornisce anche una serie di email personali che il gruppo ha deciso di utilizzare come spiegato nelle sezioni 4.1.1.1 e 4.1.1.3.

**4.1.4.5 Skype** Skype è l'applicativo scelto per effettuare videochiamate o chiamate VoIP tra i componenti del gruppo quando c'è la necessità di consultarsi o risolvere problemi e non è possibile essere presenti fisicamente nello stesso ambiente.

**4.1.4.6 WhatsApp** WhatsApp è l'applicativo di messaggistica scelto per le comunicazioni interne e informali al gruppo.

**4.1.4.7 TO DO - Strumento per la Presentazione** TO DO

## 4.2 Gestione delle Infrastrutture

### 4.2.1 Attività

**4.2.1.1 Gestione del Repository** Il gruppo ha deciso di utilizzare due repository che servono a svolgere funzioni diverse, ma necessarie, allo sviluppo del sistema finale.

Il servizio di hosting scelto consente, tramite licenza “educational”, di impostare la visibilità degli ambienti come privata.

Una volta iscritti i membri dovranno comunicare all’*Amministratore di Progetto* il loro nome utente che provvederà ad autorizzarne l’accesso.

- **doc\_BDSM\_App:** gestione della documentazione;
- **src\_BDSM\_App:** gestione del codice;

**4.2.1.2 Gestione dei Git Hooks** L’*Amministratore di Progetto* ha il compito di mantenere gli script che permettono di tenere il repository consistente con le norme descritte in questo documento.

Questo strumento per sua natura ha dei limiti, in quanto deve essere installato localmente in ogni macchina da parte dei singoli membri.

L’*Amministratore di Progetto* provvederà a notificare ai membri del gruppo quando applicare la procedura descritta nella sezione 4.2.2.1 tramite i sistemi visti nella sezione 4.1.1.1.2.

La notifica potrà avvenire principalmente in due casi:

- ogni volta che un membro clona il repository in locale nella sua macchina;
- ogni volta che l’*Amministratore di Progetto* effettua un aggiornamento degli script.

**4.2.1.3 Gestione del template del messaggio di commit** L’*Amministratore di Progetto* ha il compito di mantenere il meno ambiguo possibile l’ambiente di lavoro. È per questo che si è deciso di utilizzare un template standard per andare a scrivere il messaggio della commit. Questo strumento per sua natura ha dei limiti, in quanto deve essere installato localmente in ogni macchina da parte dei singoli membri.

L’*Amministratore di Progetto* provvederà a notificare ai membri del gruppo quando applicare la procedura descritta nella sezione 4.2.2.2 tramite i sistemi visti nella sezione 4.1.1.1.2.

La notifica potrà avvenire principalmente in due casi:

- la prima volta che un membro entra a fare parte del team di lavoro;
- ogni volta che l’*Amministratore di Progetto* effettua un aggiornamento degli script.

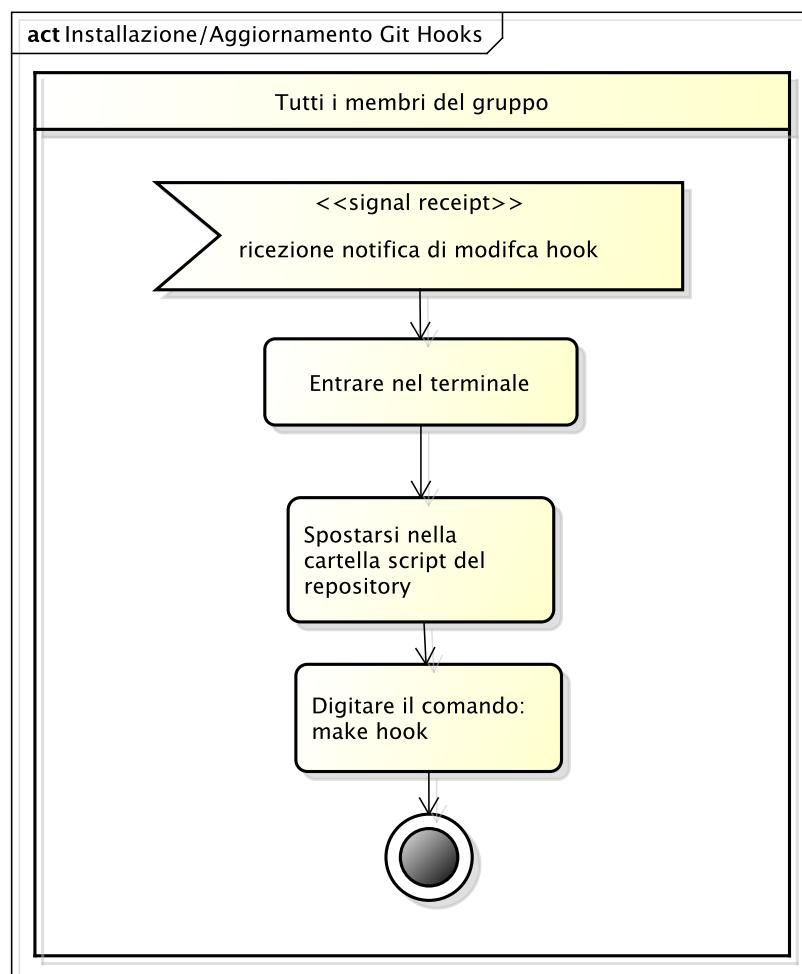
### 4.2.2 Procedure

**4.2.2.1 Installazione e manutenzione Git Hooks** I membri del team dovranno eseguire questa procedura una volta ricevuta la notifica da parte dell’*Amministratore di Progetto*.

Il diagramma presente nella figura 7 illustra in maniera più schematizzata i compiti da eseguire.

1. entrare dalla root del repository tramite terminale nella cartella “script”;
2. digitare sempre da terminale il comando:

make hook



**Figura 7:** Diagramma di attività - installazione/aggiornamento dei Git Hooks

**4.2.2.2 Installazione e manutenzione messaggio di commit** I membri del team dovranno eseguire questa procedura una volta ricevuta la notifica da parte dell’Amministratore di Progetto.

Il diagramma presente nella figura 8 illustra in maniera più schematizzata i compiti da eseguire.

1. scaricare il nuovo template allegato alla notifica ricevuta che avrà nome “git-message.txt”;
2. se è la prima volta che si esegue la procedura, lanciare il seguente comando dal terminale, altrimenti passare al passo successivo:

```
git config --global core.editor "mate"
```

Sostituite “mate” con l’editor che preferite;

3. entrare nella directory principale dell’utente (esempio: /Users/nomeutente);
4. copiare il file allegato in questa cartella rinominandolo in modo da renderlo nascosto inserendo un punto davanti al nome (esempio: .gitmessage.txt).

#### 4.2.3 Norme

##### 4.2.3.1 Repository

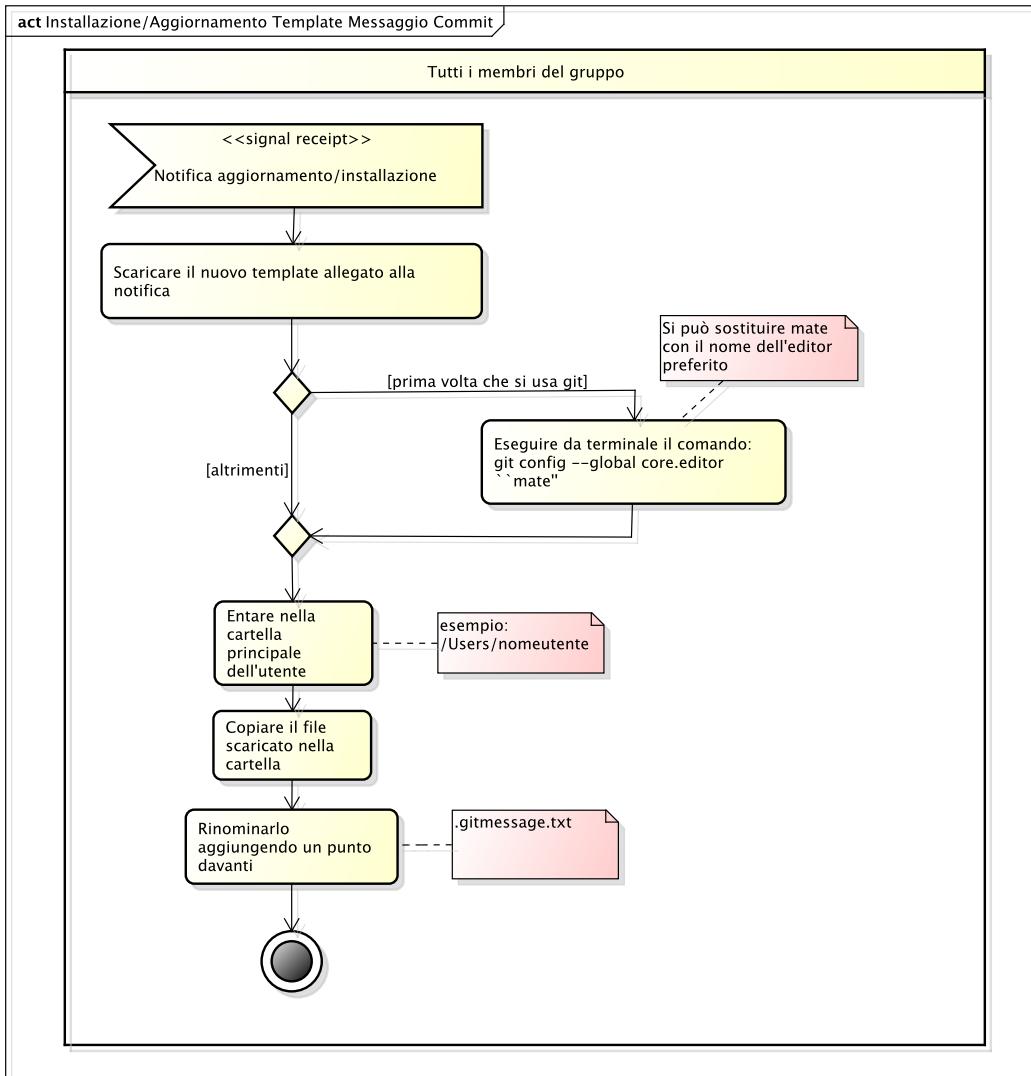
**4.2.3.1.1 Nomi dei file in doc\_BDSM\_App** I file e le cartelle presenti nel repository devono essere conformi al seguente formalismo tratto dallo Standard ISO 9660:1999 (Level 2):

- i caratteri usati sono solo quelli minuscoli a-z, 0-9, l’underscore (\_) e il punto (.) (esempio: nome\_del\_documento.tex);
- non sono ammessi caratteri accentati;
- i nomi non possono includere spazi o finire con un punto (.);
- i nomi non devono contenere più di un punto (.) ad eccezione di quelli che fanno riferimento ad una specifica versione (esempio: studio\_di\_fattibilita\_v1.0.0.pdf);
- i nomi non devono essere più lunghi di 21 caratteri esclusi i 3 destinati all’estensione.

**4.2.3.1.2 Struttura di doc\_BDSM\_App** Le cartelle nel repository verranno organizzate nel seguente modo a partire dalla root:

- **consegne:** contenente i documenti finali con anche la versione da consegnare alle diverse revisioni. Sono presenti le cartelle:
  - **revisione\_dei\_requisiti:** contenente i documenti necessari alla revisione dei requisiti;
  - **revisione\_di\_progettazione:** contenente i documenti necessari alla revisione di progettazione;
  - **revisione\_di\_qualifica:** contenente i documenti necessari alla revisione di qualifica;
  - **revisione\_di\_accettazione:** contenente i documenti necessari alla revisione di accettazione.

All’interno di ciascuna di esse ci sarà un’ulteriore suddivisione in due categorie:



**Figura 8:** Diagramma di attività - installazione/aggiornamento del template del messaggio per la commit

- **interni**: contenente i documenti necessari al gruppo per la sua organizzazione;
- **esterni**: contenente i documenti necessari alla pianificazione e all'avanzamento dello sviluppo.

- **documenti**:

- **template**: contenente i file che servono per gestire in maniera univoca la redazione di un documento;
- **template\_document**: contenente i file di esempio che dovranno essere utilizzati per ogni documento reale;
- una cartella per ogni documento che avrà come nome quello del documento in questione (esempio: norme\_di\_progetto).

- **script**: contenente tutti gli script necessari ad automatizzare il lavoro e il controllo della documentazione.

**4.2.3.1.3 Nomi dei file in src\_BDSM\_App** La presente sezione verrà redatta in futuro, presumibilmente nella prima fase di progettazione di dettaglio e codifica dei requisiti come esposto nel documento *Piano di Progetto v0.0.1*.

**4.2.3.1.4 Struttura di src\_BDSM\_App** La presente sezione verrà redatta in futuro, presumibilmente nella prima fase di progettazione di dettaglio e codifica dei requisiti come esposto nel documento *Piano di Progetto v0.0.1*.

**4.2.3.1.5 Modello di sviluppo** Per lo sviluppo della documentazione e del codice necessari al progetto si è scelto di adottare il modello proposto dal proponente, spiegato nel dettaglio al seguente link:

<http://nvie.com/posts/a-successful-git-branching-model/>

Ogni membro del gruppo dovrà leggere l'articolo e applicarlo secondo le norme di denominazione dei branch presenti in esso.

**4.2.3.1.6 Template messaggio di commit** Il messaggio di commit dovrà essere conforme alla seguente notazione:

Title:

Desc:

Task: Reported in #id\_task

Option: [option] [option]

END

- **Title**: inserire una breve descrizione come titolo di quello che è stato fatto (massimo 43 caratteri);
- **Desc**: inserire una descrizione più esaustiva dell'attività svolta qualora non fosse sufficiente quella data nel titolo;
- **Task**: al posto della dicitura “id\_task” inserire l’id del task effettivo reperibile su Asana secondo le modalità esposte nella sezione 4.1.2.2;
- **Option**: Le opzioni applicabili sono:

- **complete**: per chiudere il task qualora fosse l'ultimo necessario allo svolgimento del compito indicato;
- **delay**: per aggiungere il tag DELAY al task se si è in ritardo con lo svolgimento del compito assegnato.

#### 4.2.4 Strumenti

**4.2.4.1 Git** Git è il sistema di controllo di versione utilizzato per entrambi i repository del team.

**4.2.4.2 GitHub** GitHub è il servizio web di hosting adottato per tenere una copia del repository del progetto.

**4.2.4.3 Git Hooks** I Git Hooks sono degli script personalizzabili che vengono eseguiti in corrispondenza di un determinato evento avvenuto nel repository.

Vengono utilizzati per controllare il rispetto delle norme in maniera automatizzata non permettendo a tutto ciò che non è conforme di entrare nel sistema.

Sono anche impiegati per automatizzare la gestione dei task da parte dei componenti del gruppo secondo le norme presenti nella sezione 4.2.3.1.6.

Risiedono nella seguente cartella a partire dalla root principale del repository:

.git/hooks

**4.2.4.4 Google Drive** Google Drive è lo strumento che si è scelto di utilizzare per gestire file che non necessitano di essere sottoposti a controllo di versione. In particolare verrà impiegato per condividere manuali di utilità alla formazione dei membri del gruppo o per la stesura di idee veloci che andranno poi riviste e documentate ufficialmente nell'apposito repository.

**4.2.4.5 Sistema Operativo** I membri del gruppo operano su due sistemi operativi diversi.

Una parte utilizza Linux con distribuzione Ubuntu 14.04 e l'altra utilizza Mac OS X con versione 10.10 Yosemite.

### 4.3 Formazione dei membri del team

I membri del gruppo, per soddisfare le richieste assegnate dal *Responsabile di Progetto* al quale non sanno fare fronte con le conoscenze attuali in loro possesso, dovranno documentarsi adeguatamente durante ore esterne a quelle di lavoro, non imputabili perciò come costi al proponente.

I membri possono trovare materiale utile a questo scopo nel luogo dove risiedono i file che non necessitano di versionamento come specificato nella sezione 4.2.4.4