



Norme di Progetto

Informazioni sul documento

Nome documento	Norme di Progetto
Versione	v3.0.0
Data redazione	2013-12-02
Redattori	<ul style="list-style-type: none">• Adami Alberto• Bissacco Nicolò
Verificatori	<ul style="list-style-type: none">• Martignago Jimmy
Approvazione	<ul style="list-style-type: none">• Feltre Beatrice• <i>Seven Monkeys</i>
Lista distribuzione	<ul style="list-style-type: none">• <i>Prof. Tullio Vardanega</i>• <i>Prof. Riccardo Cardin</i>
Uso	Interno

Sommario

Documento contenente l'insieme di norme stabilite dal gruppo *Seven Monkeys* per la realizzazione di Romeo.

Diario delle Modifiche

Modifica	Autore & Ruolo	Data	Versione
<i>Inserimento automatizzazione documenti nella sottosezione Processo di documentazione</i>	Martignago Jimmy <i>Amministratore di Progetto</i>	2014-04-08	v3.0.1
<i>Approvazione del documento</i>	Scapin Davide <i>Responsabile di Progetto</i>	2014-02-17	v3.0.0
<i>Verifica del documento</i>	Adami Alberto <i>Verificatore</i>	2014-02-14	v2.1.0
<i>Riorganizzazione documento in base a direttive RP</i>	Magnabosco Nicola <i>Amministratore di Progetto</i>	2014-02-05	v2.0.1
<i>Approvazione del documento</i>	Adami Alberto <i>Responsabile di Progetto</i>	2014-02-03	v2.0.0
<i>Verifica del documento</i>	Bissacco Nicolò <i>Verificatore</i>	2014-01-31	v1.3.0
<i>Apportate modifiche a seguito della verifica</i>	Luisetto Luca <i>Amministratore di Progetto</i>	2014-01-26	v1.2.1
<i>Verifica del documento</i>	Bissacco Nicolò <i>Verificatore</i>	2014-01-24	v1.2.0
<i>Incremento sezione 7.4.4</i>	Feltre Beatrice <i>Amministratore di Progetto</i>	2014-01-20	v1.1.4
<i>Incremento sezione Ambiente di lavoro</i>	Feltre Beatrice <i>Amministratore di Progetto</i>	2014-01-14	v1.1.3
<i>Correzione degli errori rilevati nella Revisione dei Requisiti</i>	Luisetto Luca <i>Amministratore di Progetto</i>	2014-01-13	v1.1.2
<i>Cambiamenti alla struttura del documento in base ai suggerimenti del Committente</i>	Luisetto Luca <i>Amministratore di Progetto</i>	2014-01-10	v1.1.1
<i>Eseguita verifica del documento</i>	Luisetto Luca <i>Verificatore</i>	2014-01-07	v1.1.0
<i>Incremento sezione Ambiente di lavoro</i>	Feltre Beatrice <i>Amministratore di Progetto</i>	2014-12-27	v1.0.1
<i>Approvazione del documento</i>	Feltre Beatrice <i>Responsabile di Progetto</i>	2013-12-17	v1.0.0
<i>Eseguita verifica del documento</i>	Martignago Jimmy <i>Verificatore</i>	2013-12-16	v0.4.0
<i>Aggiunte modifiche a seguito verifica</i>	Adami Alberto <i>Amministratore di Progetto</i>	2013-12-15	v0.3.1
<i>Eseguita verifica del documento</i>	Martignago Jimmy <i>Verificatore</i>	2013-12-14	v0.3.0
<i>Stesura sezioni progettazione e verifica</i>	Adami Alberto <i>Amministratore di Progetto</i>	2013-12-13	v0.2.3
<i>Stesura sezione analisi dei requisiti</i>	Bissacco Nicolò <i>Amministratore di Progetto</i>	2013-12-12	v0.2.2
<i>Stesura sezione ambiente di lavoro</i>	Adami Alberto <i>Amministratore di Progetto</i>	2013-12-11	v0.2.1
<i>Eseguita verifica del documento</i>	Martignago Jimmy <i>Verificatore</i>	2013-12-07	v0.2.0

<i>Aggiunte modifiche a seguito verifica</i>	Bissacco Nicolò <i>Amministratore di Progetto</i>	2013-12-07	v0.1.1
<i>Eseguita verifica del documento</i>	Martignago Jimmy <i>Verificatore</i>	2013-12-06	v0.1.0
<i>Aggiunta sezione glossario</i>	Adami Alberto <i>Amministratore di Progetto</i>	2013-12-06	v0.0.5
<i>Stesura sezione protocollo</i>	Bissacco Nicolò <i>Amministratore di Progetto</i>	2013-12-05	v0.0.4
<i>Aggiunta delle sezioni documenti e codifica e convenzioni</i>	Adami Alberto <i>Amministratore di Progetto</i>	2013-12-04	v0.0.3
<i>Stesura delle sezioni comunicazioni e repository</i>	Bissacco Nicolò <i>Amministratore di Progetto</i>	2013-12-03	v0.0.2
<i>Inizio stesura del documento e stesura delle sezioni introduzione e incontri.</i>	Adami Alberto <i>Amministratore di Progetto</i>	2013-12-02	v0.0.1

Indice

1	Introduzione	1
1.1	Scopo del documento	1
1.2	Glossario	1
1.3	Riferimenti	1
1.3.1	Normativi	1
1.3.2	Informativi	1
2	Processi primari	2
2.1	Processo di sviluppo	2
2.1.1	Analisi dei requisiti	2
2.1.1.1	<u>Identificazione degli stakeholders:</u>	2
2.1.1.2	<u>Identificazione dei casi d'uso:</u>	2
2.1.1.3	<u>Documentazione dei casi d'uso:</u>	3
2.1.1.4	<u>Tracciamento dei casi d'uso:</u>	3
2.1.1.5	<u>Identificazione dei requisiti:</u>	3
2.1.1.6	<u>Documentazione dei requisiti:</u>	4
2.1.1.7	<u>Tracciamento dei requisiti:</u>	4
2.1.2	Progettazione architetturale	5
2.1.2.1	<u>Identificazione dei componenti:</u>	5
2.1.2.2	<u>Identificazione delle interfacce:</u>	5
2.1.2.3	<u>Documentazione dei componenti:</u>	5
2.1.2.4	<u>Tracciamento dei componenti:</u>	6
2.1.3	Progettazione architetturale di dettaglio	6
2.1.3.1	<u>Raffinamento delle componenti:</u>	6
2.1.3.2	<u>Documentazione delle classi:</u>	6
2.1.3.3	<u>Definizione test d'unità:</u>	6
2.1.4	Codifica e testing delle componenti	6
2.1.4.1	<u>Costruzione software:</u>	6
2.1.4.2	<u>Convenzioni di codifica:</u>	7
2.1.4.2.1	Nomenclatura:	7
2.1.4.2.2	Intestazione di un file:	7
2.1.4.2.3	Commenti nei metodi:	8
2.1.4.2.4	Commenti nei file di implementazione:	8
2.1.4.3	<u>Esecuzione dei test d'unità:</u>	8
3	Processi di supporto	9
3.1	Processo di documentazione	9
3.1.1	Pianificazione	9
3.1.1.1	<u>Classificazione dei documenti:</u>	9
3.1.1.1.1	Documenti informali:	9
3.1.1.1.2	Documenti formali:	9
3.1.1.2	<u>Studio di fattibilità:</u>	10
3.1.1.3	<u>Norme di progetto:</u>	10
3.1.1.4	<u>Piano di progetto:</u>	10
3.1.1.5	<u>Piano di Qualifica:</u>	10
3.1.1.6	<u>Analisi dei requisiti:</u>	10
3.1.1.7	<u>Specifiche tecnica:</u>	10
3.1.1.8	<u>Definizione di prodotto:</u>	10
3.1.1.9	<u>Glossario:</u>	10

3.1.1.10	<u>Verbali:</u>	11
3.1.1.11	<u>Manuale utente:</u>	11
3.1.1.12	<u>Procedure di gestione dei documenti:</u>	11
3.1.1.12.1	Creazione di un documento:	11
3.1.1.12.2	Avanzamento di un documento:	12
3.1.1.13	<u>Procedure di gestione del Glossario:</u>	12
3.1.1.13.1	Inserimento di un termine:	12
3.1.1.13.2	Eliminazione di un termine:	13
3.1.2	Progettazione e sviluppo	14
3.1.2.1	<u>Versionamento:</u>	14
3.1.2.2	<u>Template:</u>	14
3.1.2.3	<u>Struttura dei documenti:</u>	14
3.1.2.3.1	Prima pagina:	15
3.1.2.3.2	Diario delle modifiche:	15
3.1.2.3.3	Indici:	15
3.1.2.3.4	Formattazione di una pagina:	15
3.1.2.4	<u>Suddivisione sezioni documenti:</u>	16
3.1.2.4.1	Studio di fattibilità:	16
3.1.2.4.2	Norme di progetto:	16
3.1.2.4.3	Piano di progetto:	16
3.1.2.4.4	Piano di qualifica:	16
3.1.2.4.5	Analisi dei requisiti:	17
3.1.2.4.6	Specifica tecnica:	17
3.1.2.4.7	Definizione di prodotto:	17
3.1.2.4.8	Glossario:	17
3.1.2.4.9	Verbali:	17
3.1.2.5	<u>Norme tipografiche:</u>	18
3.1.2.5.1	Stile di testo:	18
3.1.2.5.2	Punteggiatura:	18
3.1.2.5.3	Composizione del testo:	18
3.1.2.5.4	Formati ricorrenti:	19
3.1.2.6	<u>Componenti grafiche</u>	19
3.1.2.6.1	Immagini:	20
3.1.2.6.2	Tabelle:	20
3.1.3	Produzione	20
3.1.3.1	<u>Automatizzazione dei paragrafi</u>	20
3.2	Processo di verifica	20
3.2.1	Pianificazione:	20
3.2.1.1	<u>Tecniche di analisi:</u>	20
3.2.1.1.1	Analisi statica:	21
3.2.1.1.2	Analisi dinamica:	21
3.2.2	Verifica:	21
3.2.2.1	<u>Analisi statica del codice:</u>	22
3.2.2.2	<u>Analisi dinamica del codice:</u>	22
3.2.2.3	<u>Test d'unità:</u>	22
3.2.2.4	<u>Verifica dei diagrammi:</u>	22
3.2.2.5	<u>Verifica della documentazione:</u>	23
3.2.2.6	<u>Comunicazione e risoluzione di anomalie e discrepanze</u>	23
3.3	Processo di garanzia della qualità	24
3.3.1	Qualità del prodotto	24

3.3.2	Qualità del processo	24
4	Processi organizzativi	25
4.1	Processo di gestione del progetto	25
4.1.1	Pianificazione	25
4.1.1.1	<u>Pianificare attività:</u>	25
4.1.1.2	<u>Gestione delle risorse:</u>	25
4.1.1.2.1	Rotazione dei ruoli:	25
4.1.1.3	<u>Analisi e prevenzione dei rischi:</u>	26
4.1.2	Esecuzione e controllo	26
4.1.2.1	<u>Incontri:</u>	26
4.1.2.1.1	Incontri esterni:	26
4.1.2.1.2	Incontri interni:	26
4.1.2.1.3	Richieste di incontri:	26
4.1.2.2	<u>Creazione milestone:</u>	26
4.1.2.3	<u>Creazione ticket:</u>	26
4.1.2.4	<u>Esecuzione dei compiti:</u>	28
4.1.2.5	<u>Chiusura milestone:</u>	28
4.2	Processo di organizzazione dell'infrastruttura	28
4.2.1	Definizione dell'infrastruttura	28
4.2.1.1	<u>Ambiente di sviluppo:</u>	29
4.2.1.2	<u>Controllo di versione:</u>	29
4.2.1.2.1	Repository della documentazione:	29
4.2.1.2.2	Repository del codice:	30
4.2.1.3	<u>Stesura documentazione:</u>	31
4.2.1.4	<u>Diagrammi UML:</u>	31
4.2.1.5	<u>Diagrammi di Gantt:</u>	31
4.2.1.6	<u>Codifica:</u>	31
4.2.1.7	<u>Strumenti di coordinamento:</u>	32
4.2.1.8	<u>Strumenti di comunicazione:</u>	33
4.2.1.8.1	Comunicazioni formali:	33
4.2.1.8.2	Composizione email	33
4.2.1.8.3	Comunicazioni informali:	34
4.2.1.9	<u>Strumenti per il tracciamento:</u>	34
4.2.1.10	<u>Strumenti per la verifica:</u>	34
4.2.1.11	<u>Strumenti per l'automatizzazione:</u>	34
5	Strumenti	36
5.1	Strumenti a supporto del processo di sviluppo	36
5.1.1	Git	36
5.1.1.1	<u>Installazione:</u>	36
5.1.1.2	<u>Best Practise di interazione:</u>	36
5.1.2	VMware	36
5.1.2.1	<u>Installazione:</u>	36
5.2	Strumenti a supporto del processo di documentazione	37
5.2.1	Astah	37
5.2.1.1	<u>Installazione:</u>	37
5.2.2	L ^A T _E X	38
5.2.2.1	<u>Installazione:</u>	38
5.2.3	ReqMonkeys	38

5.2.3.1	<u>Accesso al sito internet:</u>	38
5.2.3.2	<u>Inserimento di un requisito:</u>	38
5.2.3.3	<u>Modificare la fonte di un requisito:</u>	38
5.2.3.4	<u>Modificare la descrizione e/o l'id di un requisito:</u>	39
5.2.3.5	<u>Aggiungere una fonte ad un requisito:</u>	39
5.2.3.6	<u>Aggiungere un componente ad un requisito:</u>	39
5.2.3.7	<u>Eliminazione di un requisito:</u>	39
5.2.3.8	<u>Rimuovere una fonte da un requisito:</u>	40
5.2.3.9	<u>Inserimento di una fonte:</u>	40
5.2.3.10	<u>Eliminazione di una fonte:</u>	40
5.2.3.11	<u>Inserimento di un componente:</u>	40
5.2.3.12	<u>Rimuovere un componente da un requisito:</u>	40
5.2.3.13	<u>Inserire un nuovo componente:</u>	41
5.2.3.14	<u>Modificare i dettagli di un componente:</u>	41
5.2.3.15	<u>Modificare i requisiti associati ad un componente:</u>	41
5.2.3.16	<u>Eliminare un componente dal sistema:</u>	41
5.2.3.17	<u>Inserire una nuova classe:</u>	42
5.2.3.18	<u>Modificare i dettagli di una classe:</u>	42
5.2.3.19	<u>Eliminare una classe dal sistema:</u>	42
5.2.3.20	<u>Definire una relazione tra classi:</u>	43
5.2.3.21	<u>Modificare una relazione tra classi:</u>	43
5.2.3.22	<u>Eliminare una relazione tra classi:</u>	43
5.2.3.23	<u>Gestire i metodi, gli attributi e le note delle classi:</u>	44
5.2.3.24	<u>Gestire i tipi di Romeo:</u>	44
5.2.4	TeXMaker	44
5.2.4.1	<u>Installazione:</u>	44
5.2.5	Visual Paradigm	44
5.2.5.1	<u>Installazione:</u>	45
5.3	Strumenti a supporto del processo di verifica	45
5.3.1	Aspell	45
5.3.1.1	<u>Installazione:</u>	45
5.3.1.2	<u>Eseguire un controllo ortografico:</u>	45
5.3.2	Mantis	45
5.3.2.1	<u>Accesso:</u>	45
5.4	Strumenti a supporto del processo di gestione del progetto	46
5.4.1	Dropbox	46
5.4.1.1	<u>Installazione:</u>	46
5.4.2	GanttProject	46
5.4.2.1	<u>Installazione:</u>	46
5.4.3	GitHub	47
5.4.3.1	<u>Registrazione:</u>	47
5.4.3.2	<u>Creazione milestone:</u>	47
5.4.3.3	<u>Creazione ticket:</u>	48
5.4.3.4	<u>Chiusura milestone:</u>	48
5.4.4	Google Calendar	49
5.4.4.1	<u>Registrazione:</u>	49
5.4.4.2	<u>Creare un nuovo evento:</u>	49
5.4.5	Google Drive	49
5.4.5.1	<u>Registrazione:</u>	49
5.4.5.2	<u>Creazione di un nuovo documento condiviso:</u>	50

5.4.6	Scripts	50
5.4.6.1	<u>Generare la documentazione:</u>	50
5.4.6.2	<u>Eseguire controllo ortografico:</u>	50
5.4.6.3	<u>Pulire il filesystem:</u>	50
A	Lista di controllo	51

Elenco delle figure

1	Procedura di creazione di un documento	12
2	Procedura di avanzamento di un documento	13
3	Procedimento per l'inserimento di un termine.	13
4	Procedimento per l'eliminazione di un termine	14
5	Il ciclo PDCA	24
6	Modello per la gestione di una milestone	27
7	Modello di ticket per la creazione di un compito generico	28
8	Modello di ticket per la creazione di un ticket di verifica	29
9	Organizzazione file system del repository dei documenti.	30
10	Organizzazione file system del repository del codice.	31
11	Procedura per l'utilizzo del Repository	37
12	Schermata di login di Mantis	46
13	Interfaccia di GitHub per la creazione di una milestone	48
14	Interfaccia di GitHub per la creazione di un ticket.	48

1 Introduzione

1.1 Scopo del documento

Questo documento ha lo scopo di fissare le norme che tutti i membri del gruppo *Seven Monkeys* dovranno rispettare durante lo svolgimento del progetto Romeo.

Tutti i membri sono tenuti a leggere il documento e a seguire *rigorosamente* le norme ivi descritte, al fine di garantire un lavoro efficiente, efficace e per avere un'uniformità nei documenti prodotti.

Qualora ve ne sia la necessità, ogni membro del gruppo potrà contattare l'*Amministratore di Progetto* per suggerire l'aggiunta di nuove norme, oppure cambiamenti alle norme già esistenti.

L'*Amministratore di Progetto*, dopo essersi consultato con gli altri membri del gruppo, avrà la facoltà di accettare o rifiutare i suggerimenti proposti.

Le norme esposte nel documento riguarderanno tutti gli aspetti che sono inerenti allo sviluppo del prodotto software.

Il documento, in particolare, pone l'accento sui seguenti punti:

- Le interazioni tra i vari membri del gruppo e verso l'esterno;
- Le modalità di accesso al repository **G**;
- La stesura dei documenti e le varie convenzioni di scrittura utilizzate;
- Le norme utilizzate nella scrittura del codice;
- La definizione dell'ambiente di lavoro.

1.2 Glossario

Al fine di evitare ogni ambiguità e per permettere al lettore una migliore comprensione dei termini e acronimi utilizzati nei vari documenti formali, essi sono riportati nel *Glossario v3.0.0* che contiene una descrizione approfondita di tali termini e acronimi. Ogni volta che compare un termine presente nel *Glossario*, esso è marcato con una "**G**" in pedice.

1.3 Riferimenti

1.3.1 Normativi

- Qt Coding Conventions: <http://qt-project.org/wiki/Coding-Conventions>;
- Qt Coding Style: http://qt-project.org/wiki/Qt_Coding_Style.

1.3.2 Informativi

- *Piano di Progetto v3.0.0*

2 Processi primari

2.1 Processo di sviluppo

Il seguente processo contiene le attività che il gruppo *Seven Monkeys* attuerà al fine di portare a compimento lo sviluppo di un prodotto software. All'interno sono definite le attività riguardanti l'analisi dei requisiti, la stesura del design architetturale e la codifica del prodotto, l'integrazione e i test delle varie componenti software.

Il processo di sviluppo è composto dalle seguenti attività:

- **Analisi dei requisiti;**
- **Progettazione architetturale;**
- **Progettazione architetturale di dettaglio;**
- **Codifica e testing delle componenti;**

Di seguito verranno descritte le norme che regolano lo svolgimento delle varie attività.

2.1.1 Analisi dei requisiti

Lo scopo di quest'attività è di definire i requisiti del sistema che andrà a fornire dei servizi agli utenti e agli stakeholders, in un determinato ambiente. I compiti di quest'attività constano nell'identificare gli stakeholders, i loro bisogni e le interazioni che avranno con il sistema. Successivamente si dovranno mappare tali requisiti in una serie di requisiti funzionali, qualitativi, prestazionali e di vincolo che aiuteranno la stesura di una progettazione ad alto livello del software.

2.1.1.1 Identificazione degli stakeholders: quest'azione consiste nell'individuazione di singoli stakeholders o di classi di stakeholders, che hanno interessi giustificati nel seguire il ciclo di vita del software. Tra questi, sono compresi gli utenti, i proponenti del progetto, gli sviluppatori e tutte le altre entità che entreranno in contatto con il prodotto.

Nello specifico caso di questo progetto, lo studio del capitolato e la discussione con i proponenti, costituiscono le principali fonti di informazione per raggiungere lo scopo. Gli *Analisti* sono tenuti a leggere attentamente il capitolato e a contattare i proponenti per comprendere appieno il contesto d'uso in cui si andrà a collocare il prodotto.

2.1.1.2 Identificazione dei casi d'uso: quest'azione consiste nell'individuazione dei principali casi d'uso del sistema. Quest'ultimi descrivono i modi con cui il sistema può essere utilizzato e costituiscono la base da cui gli *Analisti* deriveranno una parte dei requisiti.

Per adempiere a questo compito, gli *Analisti* sono tenuti a confrontarsi con gli stakeholders, in maniera tale da comprendere di quali funzionalità necessitino. Devono inoltre contrattare con i proponenti la quantità e la tipologia di funzionalità che il sistema dovrà fornire, in base alle risorse economiche, temporali e intellettuali disponibili per la realizzazione del progetto. Tutti i casi d'uso dovranno essere tracciati, documentati ed approvati sia dagli acquirenti, sia dal *Responsabile di Progetto*.

2.1.1.3 Documentazione dei casi d'uso: nel momento in cui un nuovo caso d'uso è individuato, è necessario procedere alla stesura della sua documentazione. Ogni caso d'uso sarà quindi corredato dalle seguenti informazioni:

- **Identificativo:** ogni caso d'uso_G va tracciato con un codice. Quest'ultimo sarà composto dalle due lettere maiuscole "UC", seguite dal numero identificativo del caso d'uso_G (es. **UC2.4.6**);
- **Diagramma:** verrà utilizzato UML_G per creare i diagrammi dei caso d'uso_G, con lo scopo di semplificare la comprensione dei requisiti.
Di seguito sono riportate le convenzioni adottate nella realizzazione dei diagrammi:
 - **Standard:** verrà adottato lo standard UML_G 2.0;
 - **Lingua:** verrà adottata la lingua italiana all'interno dei diagrammi.
- **Descrizione:** una breve didascalia che descrive il caso d'uso_G;
- **Flusso principale:** verrà fornita una descrizione del flusso principale degli eventi del caso d'uso_G;
- **Scenario alternativo:** nel caso sia presente uno scenario alternativo, il caso d'uso_G dovrà essere correlato da una descrizione di tale scenario;
- **Attori:** gli attori coinvolti nel caso d'uso_G;
- **Precondizione:** la precondizione del caso d'uso;
- **Postcondizione:** la postcondizione del caso d'uso.

2.1.1.4 Tracciamento dei casi d'uso: quest'azione consiste nella memorizzazione dei casi d'uso, in maniera tale da tenerne traccia per una futura consultazione e/o rielaborazione. Inoltre, dato che i casi d'uso costituiscono una fonte di requisiti, tenerne traccia è necessario per poterli mappare.

Nello specifico, per tracciare i casi d'uso si utilizzerà l'applicativo web *ReqMonkeys*. Tale strumento consente di:

- Inserire una fonte;
- Eliminare una fonte;

Per le procedure di esecuzione delle precedenti funzionalità, si faccia riferimento alla sezione ??.

2.1.1.5 Identificazione dei requisiti: quest'azione consiste nell'identificazione dei requisiti che il sistema software deve soddisfare. Tali requisiti descrivono:

- funzionalità e capacità che il sistema dovrà avere;
- usabilità del sistema, intesa sia come estetica della GUI_G, sia come interazione uomo-macchina;
- affidabilità legata alla gestione degli errori;
- prestazioni che il sistema deve avere;
- manutenibilità ed estensibilità del prodotto;
- qualità che il sistema dovrà garantire (definita nel *Piano di Qualifica v3.0.0*).

Per svolgere quest'azione, gli *Analisti* sono tenuti a discutere con gli stakeholders, studiare in dettaglio i casi d'uso e tenere conto delle strategie di qualità definite nel *Piano di Qualifica v3.0.0*. Inoltre, è necessario assegnare una priorità, definire il tipo dei requisiti e tracciarli con le relative fonti. Infine, ogni requisito deve essere approvato sia dagli acquirenti, sia dal *Responsabile di Progetto*.

2.1.1.6 Documentazione dei requisiti: nel momento in cui un nuovo requisito è individuato, è necessario procedere alla stesura della sua documentazione. Ogni requisito dovrà essere il più completo e non ambiguo possibile. I requisiti dovranno essere classificati per tipo e priorità, utilizzando la seguente notazione:

R[importanza][tipo][codice]

dove:

- **Importanza:** può assumere i seguenti valori:
 - **0:** in questo caso si tratta di un requisito obbligatorio;
 - **1:** in questo caso si tratta di un requisito desiderabile;
 - **2:** in questo caso si tratta di un requisito opzionale.
- **Tipo:** può assumere i seguenti valori:
 - **F:** per indicare un requisito funzionale;
 - **Q:** per indicare un requisito di qualità;
 - **P:** per indicare un requisito prestazionale;
 - **V:** per indicare un requisito di vincolo.
- **Codice:** rappresenta il codice univoco di ogni requisito, il quale va indicato in forma gerarchica.

2.1.1.7 Tracciamento dei requisiti: quest'azione consiste nella memorizzazione dei requisiti, in maniera tale da tenerne traccia per poterli sottoporre all'attenzione degli stakeholder e per poterne mappare i test di soddisfazione, le componenti ad essi associate, le fonti.

Nello specifico, per tracciare i requisiti, si utilizzerà l'applicativo web *ReqMonkeys*. Tale strumento consente di:

- Inserire un requisito;
- Modificare la fonte di un requisito;
- Modificare la descrizione o l'id di un requisito;
- Aggiungere una fonte ad un requisito;
- Aggiungere un componente ad un requisito;
- Eliminare un requisito;
- Rimuovere una fonte da un requisito;
- Rimuovere un componente da un requisito.

Per le procedure di esecuzione delle precedenti funzionalità, si faccia riferimento alla sezione ??.

2.1.2 Progettazione architetturale

Lo scopo di quest'attività è di fornire un architettura software di alto livello, il cui grado di dettaglio comprende la definizione delle classi, che implementi i requisiti definiti durante l'Analisi dei requisiti. I compiti di quest'attività constano nell'individuazione delle componenti fondamentali e dei design pattern_G che serviranno a comporre lo scheletro dell'architettura. Inoltre, è necessario individuare le componenti esterne che eventualmente andranno ad interagire con il software (classi di librerie esterne, database) e di conseguenza andranno definite le relative interfacce, responsabili di comunicare con tali entità. Infine dovrà essere prodotta una documentazione ed un tracciamento delle varie componenti.

2.1.2.1 Identificazione dei componenti: quest'azione consiste nell'identificazione dei componenti che il sistema software dovrà avere. Sulla base dei requisiti identificati durante l'Analisi dei requisiti, i *Progettisti* sono tenuti a studiare le migliori strategie implementative, al fine di soddisfarli. Inoltre, devono individuare quali design pattern_G saranno più adatti al contesto, al fine di rendere il prodotto il più efficiente ed estensibile possibile. Ogni componente ed ogni design pattern_G deve essere adeguatamente documentato.

2.1.2.2 Identificazione delle interfacce: quest'azione consiste nell'identificazione delle interfacce necessarie alla comunicazione con i componenti delle entità esterne. I *Progettisti* sono tenuti a studiare i possibili componenti esterni, necessari all'implementazione di determinate funzionalità e conseguentemente devono individuare un architettura di alto livello che implementi le interfacce necessarie alla comunicazione con tali componenti. Tra le interfacce trattate in quest'azione, sono inoltre comprese le interfacce interne. Queste ultime sono responsabili della comunicazione tra le macro componenti del sistema ed anch'esse necessitano di un'architettura ad alto livello.

2.1.2.3 Documentazione dei componenti: nel momento in cui un nuovo componente è individuato, è necessario procedere alla stesura della sua documentazione. Ogni package dovrà essere quindi corredato dalle seguenti informazioni:

- **Identificativo:** ogni package va tracciato con un nome esaustivo. Il nome dovrà essere composto da una singola parola, la cui prima lettera sarà maiuscola. Nel momento in cui vengono individuati dei sotto-package, essi avranno il nome composto come segue:

Package::Sotto-package1::Sotto-package2::...

- **Diagramma:** ogni package deve essere corredato da un diagramma di package, disegnato con il linguaggio UML_G aderente allo standard 2.0;
- **Descrizione:** ogni package deve avere una descrizione esaustiva che ne espliciti gli scopi ed i contenuti;
- **Sotto-package:** ogni package deve essere corredato dalla lista dei sotto-package che eventualmente contiene;
- **Classi contenute:** ogni package deve esprimere la lista delle classi eventualmente contenute al suo interno.

A sua volta, le classi individuate devono essere corredate dalle seguenti informazioni:

- **Identificativo:** ogni classe va tracciata con un nome esaustivo. Il nome dovrà essere il più esaustivo e comprensibile possibile, in maniera da identificare velocemente la sua funzionalità;
- **Descrizione:** ogni classe deve essere corredata da una descrizione che ne espliciti gli scopi e le funzionalità che vuole implementare;
- **Sotto-classi:** ogni classe deve indicare nella propria documentazione le classi che derivano da essa.

2.1.2.4 Tracciamento dei componenti: quest'azione consiste nel memorizzare i vari componenti individuati. Inoltre, è necessario tracciarli con i requisiti che intendono soddisfare e con i test di integrazione che andranno a verificarne la corretta interazione. Per questo, i *Progettisti* sono tenuti ad utilizzare l'applicativo **ReqMonkeys** per le varie operazioni di tracciamento, tra le quali:

- Inserimento di un nuovo componente;
- Aggiungere la dipendenza di un componente ad un requisito;
- Eliminare la dipendenza di un componente da un requisito;

Per le procedure di esecuzione delle precedenti funzionalità, si faccia riferimento alla sezione ??.

2.1.3 Progettazione architetturale di dettaglio

Lo scopo di quest'attività è di fornire un architettura software dettagliata, in maniera da soddisfare i requisiti identificati. Tale architettura deve essere sufficientemente dettagliata per permettere la codifica e la stesura dei relativi test d'unità. I compiti di quest'attività constano nell'individuazione dei metodi che le classi definite durante la progettazione ad alto livello devono implementare, nella documentazione di tali funzionalità e nella stesura dei relativi test d'unità.

2.1.3.1 Raffinamento delle componenti: quest'azione consiste nel raffinamento delle classi individuate durante la progettazione architetturale, con lo scopo di definire esaurientemente le funzionalità che ognuna deve offrire. In particolare, i *Progettisti* sono tenuti ad individuare i metodi che le classi devono implementare.

2.1.3.2 Documentazione delle classi: nel momento in cui una classe viene definita nella sua interezza, deve essere documentata. Per questo, ogni classe dovrà essere corredata dalle seguenti informazioni:

- **Diagramma:** un diagramma UML_G che rappresenta la classe;
- **Descrizione:** una breve descrizione del funzionamento della classe in questione;
- **Utilizzo:** il contesto di utilizzo della classe all'interno del software;
- **Attributi:** verranno indicati i vari attributi della classe, corredata da una breve descrizione;
- **Metodi:** verranno indicati i metodi della classe, corredata dalla propria segnatura e da una breve descrizione.

2.1.3.3 Definizione test d'unità: i *Progettisti* dovranno provvedere alla creazione di test d'unità per verificare il corretto funzionamento delle varie componenti del sistema.

2.1.4 Codifica e testing delle componenti

Lo scopo di quest'attività è di costruire delle unità software, in accordo con la specifica prodotta dalla progettazione architetturale di dettaglio. Tali unità dovranno soddisfare i requisiti della specifica e dovranno essere compilate e testate. Dovrà essere inoltre fornita una documentazione dei risultati dei test d'unità effettuati.

2.1.4.1 Costruzione software: quest'azione consiste nella codifica e nella documentazione delle unità software e delle unità del database. I *Programmatori* sono tenuti a costruire le unità software seguendo le specifiche della definizione di prodotto.

2.1.4.2 Convenzioni di codifica: di seguito sono riportate le convenzioni che i *Programmatori* dovranno seguire durante la fase di Codifica.

Per rendere il codice più leggibile e facilmente comprensibile, i *Programmatori* dovranno seguire le QtG Coding Convention e le QtG Styling Code. (Vedi sezione 1.3.1)

È consentita la possibilità di effettuare dei cambiamenti alle convenzioni stabilite, in seguito ad una decisione del *Responsabile di Progetto*.

2.1.4.2.1 Nomenclatura: per l'assegnazione di nomi a classi, variabili, metodi e costanti andranno seguite le seguenti regole:

- **Classi:** va utilizzata la notazione mixed case, con la prima lettera maiuscola;
- **Metodi:** va utilizzata la notazione mixed case, con la prima lettera minuscola;
- **Variabili:** va utilizzata la notazione mixed case, con la prima lettera minuscola;
- **Costanti:** va scritto il nome *interamente* in maiuscolo, separando le varie parole con il carattere “_” (underscore).

2.1.4.2.2 Intestazione di un file: ad ogni file di codice dovrà corrispondere una singola classe propria.

Una classe dovrà essere suddivisa in un file *header* (.h) e in un file di *implementazione* (.cpp). Ogni file *header* dovrà necessariamente avere un'intestazione, la quale dovrà avere la seguente forma:

```

/#!
*file Nome del file
*author Cognome Nome (e-mail)
*date Data di creazione del file
*brieff Descrizione breve della classe
*Descrizione dettagliata della classe
*/
/*
*Changes:
*+-----+-----+-----+-----+-----+
*| Version + Date + Programmer +          Changes          + Description |
*+-----+-----+-----+-----+-----+
*|
*|  x.y.z  | AAMGG | Nome Cognome | ClassName::MethodName | Description |
*|
*+-----+-----+-----+-----+-----+
*/

```

dove:

- **File:** deve essere il nome del file, comprendente di estensione (.h o .cpp);
- **Author:** deve essere il creatore del file e non necessariamente il programmatore che sta modificando il file attualmente;
- **Date:** deve essere la data di creazione del file, espressa come indicato in sezione 3.1.2.5.4;
- **Brieff:** è una descrizione breve della classe (massimo due righe);
- **Changes:** rappresenta la tabella di avanzamento del file, comprensiva di modifiche, aggiunte ed eliminazioni di metodi della classe.

Nello specifico la tabella deve avere le seguenti righe:

- **Version:** la versione del file successivamente alla modifica;
- **Date:** la data in cui è stato modificato il file;

- **Programmer:** il programmatore che ha fatto la modifica, che può non essere necessariamente il creatore del file;
- **Changes:** rappresenta il metodo che ha subito cambiamenti;
- **Description:** una breve descrizione del cambiamento effettuato.

2.1.4.2.3 Commenti nei metodi: sempre nel file *header* prima di ogni metodo, si dovrà inserire un commento che dovrà rispettare la seguente forma:

```
/*!brief Breve descrizione del metodo
*Descrizione dettagliata del metodo.
*\param Descrizione del primo parametro
*\param Descrizione dell'n-esimo parametro
*\return il tipo del valore di ritorno
*/
/*
*Pre-Condition: {La pre-condizione}
*Post-Condizione: {La post-condizione}
*/
```

dove:

- **Brief:** deve essere una breve descrizione del funzionamento del metodo (massimo due righe);
- **Pre-Condition:** deve essere la pre condizione che vale prima della chiamata del metodo;
- **Post-Condition:** deve essere la post condizione che varrà dopo la chiamata del metodo;
- **Return:** deve essere il tipo di ritorno;

2.1.4.2.4 Commenti nei file di implementazione: nei vari file di implementazione è gradita la presenza di commenti in modo da facilitare la comprensione del codice. I commenti devono essere scritti nel modo più chiaro e descrittivo possibile. Ogni variabile di particolare importanza, o il quale utilizzo è particolarmente complesso, dovrà essere accompagnata da una breve descrizione del suo scopo. In particolare è gradita la presenza di *invarianti*, *pre* e *post condizioni* sui cicli, nel caso in cui essi abbiano almeno due livelli di annidamento.

2.1.4.3 Esecuzione dei test d'unità: quest'azione consiste nell'esecuzione dei test d'unità, in maniera tale da assicurare che soddisfino i requisiti definiti. I risultati devono essere documentati.

3 Processi di supporto

3.1 Processo di documentazione

Il processo di documentazione è un processo che consente di registrare le informazioni prodotte dalle attività svolte durante il ciclo di vita del software. Tale processo vuole individuare una strategia per lo sviluppo della documentazione. In particolare, in questa sezione, verranno individuati quali documenti sono necessari come supporto alle attività di sviluppo e verranno presentati gli standard che il gruppo *Seven Monkeys* intende adottare per la loro produzione.

Il processo di documentazione è composto dalle seguenti attività:

- **Pianificazione;**
- **Progettazione e sviluppo;**
- **Produzione;**
- **Manutenzione;**

3.1.1 Pianificazione

Lo scopo di quest'attività è di identificare quali documenti necessitino di essere prodotti durante il ciclo di vita del software. Per ogni documento, dovrà esserne definito: il titolo, lo scopo, i destinatari e le procedure di gestione.

3.1.1.1 Classificazione dei documenti: in questo paragrafo verranno definiti i tipi di documenti che si dovranno produrre. Verranno inoltre fornite le norme su come denominarne i file.

3.1.1.1.1 Documenti informali: si definiscono documenti informali tutti quei documenti che sono ancora in fase di stesura e devono ancora essere approvati dal *Responsabile di Progetto*.

Tali documenti sono ad *esclusivo* uso interno e non potranno essere divulgati a terze parti, prima di essere stati verificati ed approvati.

Una volta approvati, i documenti diventeranno formali e pronti per la distribuzione verso l'esterno.

Tutti i documenti informali dovranno essere salvati nel repository^G nell'apposita cartella¹.

Tali documenti devono essere rinominati osservando le seguenti regole:

- La prima lettera di ogni parola, che non sia una preposizione, deve essere necessariamente maiuscola;
- Gli spazi devono essere sostituiti con il carattere “_” (underscore);
- I caratteri accentati devono essere sostituiti con i medesimi caratteri, ma privi di accento.

Un esempio di nome è il seguente: *Studio_di_Fattibilita.pdf*.

3.1.1.1.2 Documenti formali: si definiscono documenti formali tutti i documenti che sono stati approvati dal *Responsabile di Progetto*² e sono pronti per essere rilasciati.

Quando un documento diventa formale potrà essere visionato da terze parti.

Il file pronto per il rilascio dovrà seguire le stesse norme di nomenclatura del file in fase di

¹Per maggiori dettagli vedere la sezione 4.2.1.2.1.

²Fatta eccezione per il *Piano di Progetto*, che viene approvato da un qualsiasi altro membro del gruppo.

stesura con l'aggiunta, alla fine del nome, della versione separata dal carattere "-" (trattino) e preceduta da una "v" (es. *Studio_di_Fattibilita-v1.0.0.pdf*).

3.1.1.2 Studio di fattibilità: lo scopo di tale documento è di riassumere le considerazioni che portano il gruppo *Seven Monkeys* all'accettazione o meno di una proposta per lo sviluppo di un progetto. Nel momento in cui si debba decidere tra svariate proposte, è consentita la stesura di un documento riassuntivo contenente le considerazioni di tutte le proposte.

L'uso del documento è interno e la lista di distribuzione comprende solo i committenti.

3.1.1.3 Norme di progetto: lo scopo di tale documento è di raccogliere le norme, le procedure e gli strumenti di sviluppo che il gruppo *Seven Monkeys* intende adottare nel momento in cui venga accettata la produzione di un nuovo progetto.

L'uso del documento è interno e la lista di distribuzione comprende solo i committenti.

3.1.1.4 Piano di progetto: lo scopo di tale documento è di specificare la pianificazione dei lavori durante lo svolgimento di un progetto, pianificare le tempistiche di attuazione delle attività, preventivare e mettere a consuntivo le risorse necessarie ed infine definire una politica di gestione dei rischi.

L'uso del documento è esterno e la lista di distribuzione comprende i committenti ed i proponenti.

3.1.1.5 Piano di Qualifica: lo scopo di tale documento è di descrivere le strategie che il team intende adottare, al fine di perseguire gli obiettivi di qualità che si vogliono ottenere all'eventuale progetto in corso. In questo documento, saranno inoltre presenti i resoconti dell'attività di verifica e le pianificazioni dei vari test che si andranno ad implementare.

L'uso del documento è esterno e la lista di distribuzione comprende i committenti ed i proponenti.

3.1.1.6 Analisi dei requisiti: lo scopo di tale documento è di fornire un prospetto dell'analisi che verrà effettuata sulla proposta di sviluppo di un progetto, al fine di individuare i requisiti. Il documento conterrà la lista dei casi d'uso e dei requisiti individuati, oltre ai diagrammi delle attività riguardanti le interazioni tra utenti e sistema.

L'uso del documento è esterno e la lista di distribuzione comprende i committenti ed i proponenti.

3.1.1.7 Specifica tecnica: lo scopo di tale documento è di definire la progettazione ad alto livello del sistema che il gruppo *Seven Monkeys* ha eventualmente deciso di sviluppare. Il documento conterrà la descrizione delle varie componenti necessarie al sistema, individuate dai *Progettisti*, corredate dai relativi design pattern^G utilizzati.

L'uso del documento è esterno e la lista di distribuzione comprende i committenti ed i proponenti.

3.1.1.8 Definizione di prodotto: lo scopo di tale documento è di definire la progettazione di dettaglio del sistema che il gruppo *Seven Monkeys* ha eventualmente deciso di sviluppare. Il documento avrà un grado di dettaglio tale da permettere ai *Programmatore* di poter codificare le varie unità software.

L'uso del documento è esterno e la lista di distribuzione comprende i committenti ed i proponenti.

3.1.1.9 Glossario: lo scopo di tale documento è di raccogliere i termini particolari presenti nei vari documenti, che necessitano di un approfondimento. In questi sono presenti

gli acronimi e i termini tecnici non immediatamente comprensibili ai lettori della documentazione.

L'uso del documento è esterno e la lista di distribuzione comprende i committenti ed i proponenti.

3.1.1.10 Verbalì: lo scopo di tali documenti è di raccogliere le informazioni salienti che emergono dalle riunioni esterne ed interne del gruppo. Le decisioni che verranno prese durante le riunioni dovranno essere tracciate.

Tutti i verbali esterni saranno redatti dal *Responsabile di Progetto*, mentre i verbali interni saranno redatti da un qualsiasi membro presente all'incontro.

I verbali dovranno essere denominati secondo il seguente criterio:

Verbale{numero del verbale}_{tipo di incontro}_{data incontro}

dove:

- **Numero del verbale:** indica un numero progressivo che identifica il verbale³;
- **Tipo di incontro:** indica il tipo di incontro avvenuto, che potrà essere:
 - **Interno:** nel caso si tratti di un incontro interno;
 - **Esterno:** nel caso si tratti di un incontro esterno.
- **Data incontro:** indica la data in cui è avvenuto l'incontro⁴.

L'uso del documento è interno, quando si riferisce ad una riunione interna, viceversa è ad uso esterno. La lista di distribuzione comprende sempre sia committenti che proponenti.

3.1.1.11 Manuale utente: lo scopo di tale documento è di fornire all'utente finale la descrizione dettagliata di tutte le funzionalità che il prodotto offrirà.

3.1.1.12 Procedure di gestione dei documenti: nel seguente paragrafo sono riportate l'elenco di procedure e regole seguite dal gruppo *Seven Monkeys* per la gestione dei documenti.

3.1.1.12.1 Creazione di un documento: la creazione di un nuovo documento è a discrezione esclusiva del *Responsabile di Progetto*. Nel momento in cui sia necessario creare un nuovo documento, egli dovrà seguire la seguente procedura (fig. 1):

1. Creare una nuova cartella all'interno di Documenti/ e denominarla con lo stesso nome del documento che si vuole creare;
2. Copiare la cartella *doc_to_modify* (contenuta nella cartella *Template/*) ed il suo contenuto, all'interno della cartella creata al punto 1;
3. Creare una cartella *Content* all'interno della cartella creata al punto 1. Questa directory conterrà i file di compilazione corrispondenti alle varie sezioni del documento;
4. Rinominare il file *template.tex* presente nella cartella *doc_to_modify*, utilizzando lo stesso nome del documento che si vuole creare. Successivamente, spostarlo all'interno della cartella creata al punto 1;
5. Compilare i campi dei file *template* precedentemente copiati, in base al documento che si vuole creare;
6. Creare i vari file di compilazione corrispondenti alle sezioni del documento all'interno della cartella *Content*.

³Tale numero dovrà partire da uno.

⁴La data dovrà essere formattata come indicato in sezione 3.1.2.5.4.

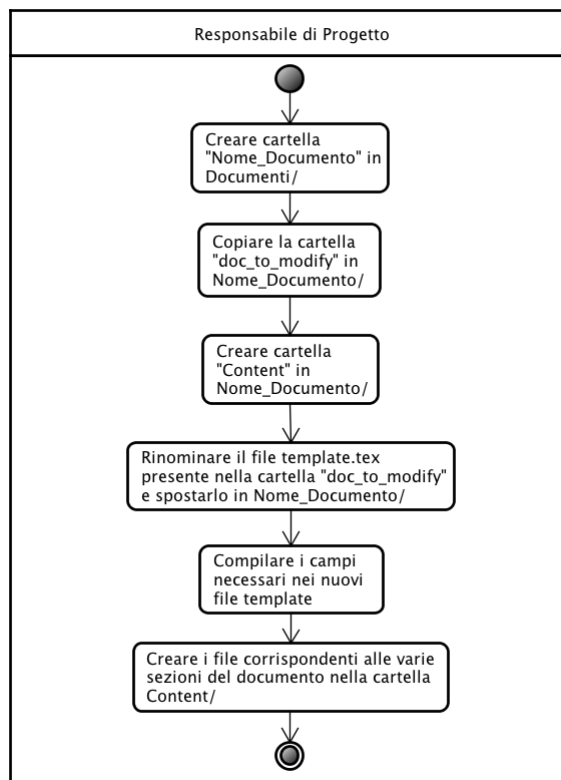


Figura 1: Procedura di creazione di un documento

3.1.1.12.2 Avanzamento di un documento: ogni qualvolta un redattore ritiene che la stesura di un documento sia terminata andrà seguita la seguente procedura (fig. 2):

1. Il redattore dovrà contattare il *Responsabile di Progetto* per informarlo della terminazione della redazione del documento;
2. Il *Responsabile di Progetto* provvederà ad assegnare la verifica del documento ad uno dei *Verificatori* disponibili che non sia in conflitto d'interessi;
3. Il *Verificatore* provvederà alla verifica del documento;
4. Se sono stati trovati errori, il *Verificatore* provvederà alla creazione di un ticket di correzione. Successivamente il redattore correggerà gli errori trovati e infine si torna al passo uno;
5. Altrimenti il documento verrà approvato dal *Responsabile di Progetto*.

3.1.1.13 Procedure di gestione del Glossario: di seguito verranno elencate le procedure specifiche per la gestione del Glossario.

3.1.1.13.1 Inserimento di un termine: per l'inserimento di un termine nel *Glossario* si dovrà seguire la seguente procedura (fig 3):

1. Prima dell'inserimento di un termine andrà contattato il *Responsabile di Progetto* per proporre l'inserimento del termine;
2. Solo nel caso in cui il *Responsabile di Progetto* approvi l'inserimento del termine, esso andrà inserito nel *Glossario* nell'ordine lessicografico corretto;
3. Successivamente all'inserimento, si dovrà provvedere a evidenziare ogni occorrenza del termine nei vari documenti.

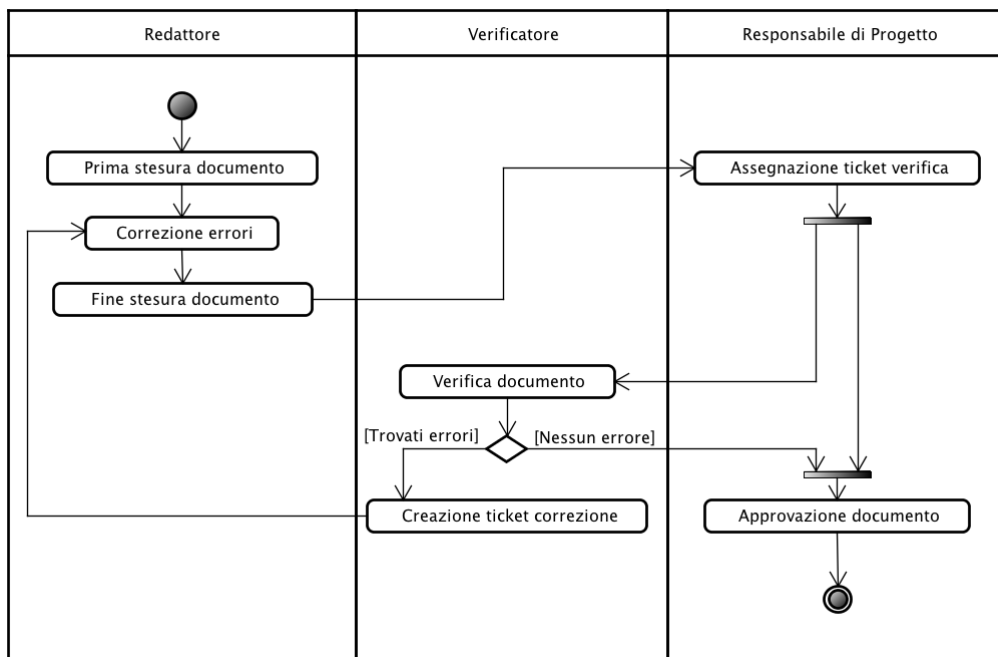


Figura 2: Procedura di avanzamento di un documento

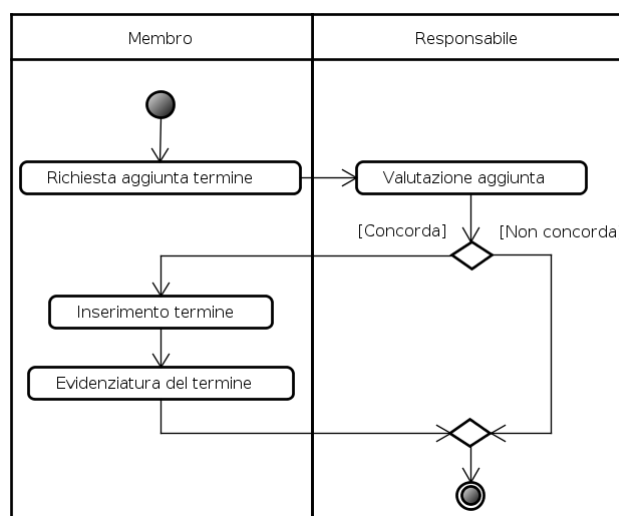


Figura 3: Procedimento per l'inserimento di un termine.

3.1.1.13.2 Eliminazione di un termine: ogni membro del gruppo per richiedere l'eliminazione di un termine del *Glossario* dovrà seguire la seguente procedura (fig. 4):

1. Dovrà essere contattato il *Responsabile di Progetto* per proporre l'eliminazione;
2. Nel caso in cui il *Responsabile di Progetto* approvi, il termine andrà cancellato dal *Glossario*;
3. Successivamente si provvederà a togliere tutte le marcature del termine, all'interno di tutti i documenti.

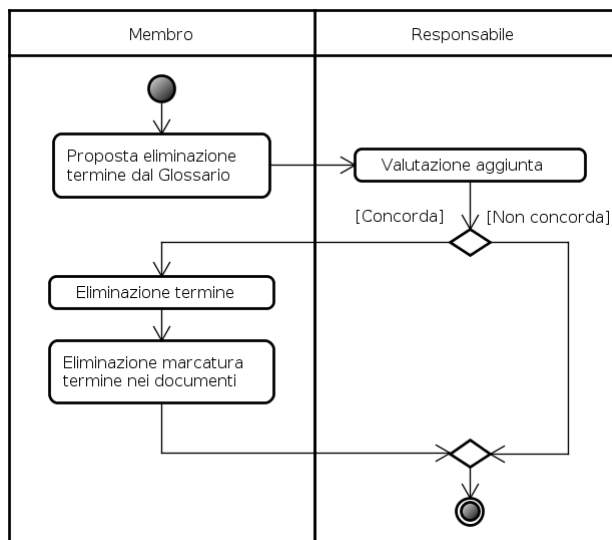


Figura 4: Procedimento per l'eliminazione di un termine

3.1.2 Progettazione e sviluppo

Lo scopo di quest'attività è di definire la struttura del layout dei documenti, le informazioni che devono essere inserite nelle prime pagine, le norme di versionamento, la composizione delle sezioni da inserire e le norme tipografiche a cui si dovranno attenere i redattori.

3.1.2.1 Versionamento: ogni documento deve specificare la propria versione, la quale sarà della seguente forma:

$X.Y.Z$

dove:

- **X:** numero che identifica la versione di rilascio. Ogni incremento causa l'azzeramento dei valori Y e Z. In particolare, questo parametro verrà incrementato ogni qual volta si consegneranno i documenti per una revisione;
- **Y:** numero che indica l'avvenuta verifica del documento. Nel momento in cui il verificatore ha controllato il documento, sarà tenuto ad incrementare questo parametro. Ogni incremento inoltre, comporta l'azzeramento del parametro Z;
- **Z:** numero che identifica una semplice modifica del documento. Ogni aggiunta o correzione al documento, comporta un incremento del parametro Z.

3.1.2.2 Template: per facilitare la stesura e la manutenzione dei vari documenti, i file di compilazione sono stati strutturati in maniera tale che la composizione generale sia gestita da template validi per ogni documento. In particolare sono stati creati template per il frontespizio, per il layout, per i comandi utilizzati nei file e per la tabella delle modifiche. Ogni documento avrà una cartella interna denominata *Content*, in cui verranno collocati i file corrispondenti alle varie sezioni che lo compongono.

3.1.2.3 Struttura dei documenti: di seguito vengono elencate le norme riguardanti la strutturazione e la suddivisione in sezioni dei vari documenti.

3.1.2.3.1 Prima pagina: ogni documento deve avere una prima pagina contenente le seguenti informazioni:

- Nome del progetto;
- Logo del gruppo;
- Nome del gruppo;
- E-mail esterna del gruppo;
- Nome del documento;
- La versione del documento⁵;
- Data di redazione del documento;
- Cognome e nome dei redattori del documento;
- Cognome e nome dei verificatori del documento;
- Cognome e nome di chi ha approvato il documento;
- Tipo d'uso del documento⁶;
- La lista di distribuzione del documento;
- Un sommario, contenente una breve descrizione del documento.

3.1.2.3.2 Diario delle modifiche: nella seconda pagina di ogni documento deve essere presente il diario delle modifiche.

Ogni riga del diario delle modifiche deve contenere le seguenti informazioni:

- Una breve descrizione sulle modifiche effettuate;
- Il cognome e nome di chi ha effettuato la modifica;
- Il ruolo dell'autore della modifica;
- La data in cui è stato modificato il documento⁷;
- La versione del documento dopo la modifica.

3.1.2.3.3 Indici: in ogni documento, tranne che nel *Glossario*, deve essere presente un indice delle sezioni. Nel caso in cui il documento contenga immagini e/o tabelle, devono essere presenti anche i relativi indici.

3.1.2.3.4 Formattazione di una pagina: l'intestazione di ogni pagina deve avere le seguenti informazioni:

- Il logo del gruppo;
- Nome del gruppo;
- La sezione corrente all'interno del documento.

A piè di pagina deve invece esserci:

- Nome del documento;
- La versione corrente del documento;
- Numero di pagina nel formato “**Pagina X di N**”, dove “X” è il numero della pagina corrente e “N” è il numero di pagine totali del documento.

⁵Espressa come indicato nel paragrafo *Versionamento* della sezione 3.1.2.

⁶Potrà essere interno o esterno.

⁷La data deve essere espressa come riportato nel paragrafo *Formati ricorrenti* in sezione 3.1.2.

3.1.2.4 Suddivisione sezioni documenti: in questo paragrafo verranno definite le principali sezioni che dovranno essere inserite nei documenti e i contenuti che ogni sezione dovrà avere. Per ogni documento saranno descritte sommariamente le sezioni al netto di frontespizio, diario delle modifiche, indice ed introduzione, in quanto sono sezioni comuni a tutti i documenti.

3.1.2.4.1 Studio di fattibilità: per ogni proposta di progetto dovranno essere presenti le seguenti sezioni:

- **Descrizione del capitolato:** contiene una descrizione sommaria del progetto;
- **Studio del dominio:** diviso in dominio applicativo e tecnologico, contiene le considerazioni riguardanti il contesto in cui si colloca il prodotto da sviluppare;
- **Valutazione complessiva:** diviso in aspetti positivi e negativi, contiene le motivazioni per cui sarebbe fattibile o meno intraprendere lo sviluppo del progetto presentato.

3.1.2.4.2 Norme di progetto: questo documento dovrà essere suddiviso nelle seguenti sezioni:

- **Processi primari:** contenente le norme riguardanti i processi primari implementati dal team;
- **Processi di supporto:** contenente le norme riguardanti i processi di supporto implementati dal team;
- **Processi organizzativi:** contenente le norme riguardanti i processi organizzativi implementati dal team;
- **Strumenti:** contenente la lista degli strumenti che il team utilizzerà durante lo sviluppo del progetto. In questa sezione verranno inoltre descritte le principali procedure di interazione che i componenti del gruppo avranno con tali strumenti.

3.1.2.4.3 Piano di progetto: questo documento dovrà essere suddiviso nelle seguenti sezioni:

- **Pianificazione:** contenente l'analisi dei rischi del progetto, la pianificazione dello svolgimento delle attività individuate e la suddivisione delle ore necessarie ad ogni ruolo e per ogni attività;
- **Suddivisione del lavoro e proposta economica:** contenente la suddivisione dei ruoli in base alle risorse disponibili ed il prospetto economico delle ore preventivate per i ruoli, in ogni fase;
- **Consuntivo corrente:** contenente il bilancio corrente tra preventivo e consuntivo;
- **Organigramma:** contenente le informazioni del team.

3.1.2.4.4 Piano di qualifica: questo documento dovrà essere suddiviso nelle seguenti sezioni:

- **Visione generale della strategia di verifica:** contiene la definizione degli obiettivi di qualità di processo e prodotto, la pianificazione delle attività di verifica in base alle risorse disponibili e la definizione delle metriche impiegate durante tale attività;
- **Gestione amministrativa della revisione:** contiene la strategia di gestione delle procedure che garantiscono la qualità di processo e prodotto;
- **Resoconto delle attività di verifica:** appendice contenente gli esiti delle attività di verifica effettuate durante le fasi del progetto;
- **Pianificazione dei test:** appendice contenente il tracciamento dei test effettuati sul prodotto software;

3.1.2.4.5 Analisi dei requisiti: questo documento dovrà essere suddiviso nelle seguenti sezioni:

- **Descrizione generale:** contiene una descrizione mediamente dettagliata del contesto d'uso del prodotto, delle funzioni che esso dovrà avere e delle caratteristiche generali degli utenti finali;
- **Casi d'uso:** contiene l'elenco dei casi d'uso individuati durante l'analisi dei requisiti, corredati dalle relative informazioni;
- **Diagrammi delle attività:** contiene una serie di diagrammi di attività che esplicheranno le possibili interazioni che gli utenti potranno avere con il prodotto finito;
- **Requisiti:** contiene la lista dei requisiti individuati, corredati da una serie di tabelle che riassumeranno il tracciamento con le fonti e con i test di validazione.

3.1.2.4.6 Specifica tecnica: questo documento dovrà essere suddiviso nelle seguenti sezioni:

- **Tecnologie utilizzate:** contiene una panoramica delle tecnologie che verranno utilizzate nell'eventuale sviluppo di un progetto;
- **Descrizione architettura:** contiene una panoramica generale dell'architettura che si intende implementare in un eventuale sviluppo di progetto;
- **Componenti e classi:** contiene un elenco strutturato che descrive in dettaglio le varie componenti che si intendono implementare nell'architettura del sistema software;
- **Design pattern:** contiene la specifica d'uso dei design pattern⁸ che si intendono utilizzare all'interno dell'architettura software;
- **Stime di fattibilità e risorse necessarie:** contiene il resoconto delle stime di fattibilità e delle risorse necessarie all'implementazione dell'architettura;
- **Tracciamento:** contiene le tabelle di tracciamento tra requisiti e componenti;
- **Descrizione dei design pattern:** appendice contenente una descrizione generale dei design pattern⁸ utilizzati nell'architettura.

3.1.2.4.7 Definizione di prodotto: questo documento dovrà essere suddiviso nelle seguenti sezioni:

- **Specifica componenti:** contiene le descrizioni dettagliate delle classi di cui è composto il software.

3.1.2.4.8 Glossario: questo documento dovrà essere suddiviso raggruppando i termini in ordine alfabetico.

3.1.2.4.9 Verballi: la prima pagina di ogni verbale deve contenere le seguenti informazioni:

- **Data**⁸;
- **Luogo ritrovo:** sintetica descrizione del luogo, correlata da indirizzo e città;
- **Ora inizio-fine:** espresso nel seguente formato ora inizio-fine: hh:mm-hh:mm;
- **Membri assenti:** i membri assenti alla riunione.

⁸Formato espresso come in paragrafo *Formati ricorrenti* della sezione 3.1.2

3.1.2.5 Norme tipografiche: al fine di evitare incoerenza tra le diverse parti dei documenti, si rimanda a questa sottosezione per tutte le informazioni riguardanti l'ortografia, la tipografia e l'assunzione di uno stile uniforme in tutti i documenti.

3.1.2.5.1 Stile di testo:

- **Grassetto:** il grassetto si deve utilizzare nei seguenti casi:
 - **Elenchi puntati:** per evidenziare l'oggetto trattato nel paragrafo;
 - **Altri casi:** è inoltre possibile utilizzare il grassetto per evidenziare termini particolarmente rilevanti.
- **Corsivo:** il corsivo deve essere utilizzato nei seguenti casi:
 - **Ruoli:** ogni riferimento a un ruolo deve essere scritto in corsivo (esempio: *Responsabile di Progetto*);
 - **Citazioni:** ogni citazione ad una fonte esterna va fatta tramite l'uso del corsivo;
 - **Documenti:** ogni riferimento a un documento deve essere scritto in corsivo (esempio: *Glossario*);
 - **File e directory:** ogni trascrizione del nome di un file o di una directory, deve essere fatta usando il corsivo;
 - **Altri casi:** in altri casi può essere necessario usare il corsivo, come per evidenziare termini particolarmente significativi.
- **Maiuscolo:** l'uso del maiuscolo è strettamente limitato alla trascrizione di acronimi;
- **L^AT_EX:** ogni riferimento a L^AT_EX va fatto tramite il comando `\LaTeX`.

3.1.2.5.2 Punteggiatura:

- **Punteggiatura:** qualsiasi segno di punteggiatura non può seguire un carattere di spazio;
- **Punti ellittici:** i punti di sospensione devono essere inseriti esclusivamente tramite il comando `LATEX \dots`, immediatamente dopo l'ultimo carattere non di spaziatura;
- **Parentesi:** il periodo racchiuso tra le parentesi non deve mai iniziare con un carattere di spazio e non deve mai terminare con un carattere di spazio o punteggiatura;
- **Lettere maiuscole:** le lettere maiuscole vanno poste dopo il punto, il punto esclamativo, il punto interrogativo e all'inizio di ogni elemento di un elenco puntato. Inoltre viene utilizzata la lettera maiuscola per i ruoli di progetto, i nomi dei documenti, le fasi di progetto, le revisioni di progetto, oltre che dove imposto dalla lingua italiana.

3.1.2.5.3 Composizione del testo:

- **Elenchi:** la prima parola che segue l'oggetto di indentazione deve avere la prima lettera minuscola. Ogni punto dell'elenco deve terminare con un carattere di punto e virgola, tranne l'ultimo che termina con un punto;
- **Note a piè di pagina:** ogni nota a piè di pagina deve iniziare con la prima lettera della prima parola maiuscola non preceduta da caratteri di spazio. Ogni nota a piè di pagina deve terminare con un punto;
- **Pedice “G”:** il pedice_G viene utilizzata in corrispondenza di termini tecnici e acronimi presenti nei documenti *Glossario*⁹.

⁹Le occorrenze dei termini presenti nei titoli, nelle note a piè di pagina e negli oggetti degli elenchi non andranno evidenziati.

3.1.2.5.4 Formati ricorrenti:

- **Nomi propri:** l'utilizzo dei nomi propri deve seguire la forma "Cognome Nome";
- **Path:** per inserire indirizzi web o indirizzi mail, deve essere utilizzato *esclusivamente* il comando $\text{\LaTeX}\ \text{\url}$;
- **Date:** ogni data deve essere espressa seguendo lo standard ISO 8601¹⁰:

$$AAAA - MM - GG$$

dove:

- **AAAA:** rappresenta il formato dell'anno scritto con tutte e quattro le cifre;
- **MM:** rappresenta il mese scritto con esattamente due cifre¹¹;
- **GG:** rappresenta il giorno scritto con esattamente due cifre.¹²
- **Riferimenti ai documenti:** ci si riferirà ai vari documenti scrivendoli in corsivo e mettendo una lettera maiuscola per ogni parola che non sia un articolo (ad esempio *Norme di Progetto*).
Nel caso ci sia la necessità di riferirsi ad una versione specifica del documento, essa andrà indicata (esempio: *Norme di Progetto v3.0.0*);
- **Nomi propri:** i nomi propri di persona devono essere scritti nel formato "Cognome Nome";
- **Nome del gruppo:** ci si riferirà al gruppo Seven Monkeys con la dicitura "*Seven Monkeys*", con il nome del gruppo in corsivo. Per la corretta scrittura è stato creato un comando \LaTeX apposito \authorName ;
- **Nome del proponente:** ci si riferirà al proponente con "Department of Information Engineering". Per la corretta scrittura è stato creato il comando \LaTeX \proposerName ;
- **Nome del progetto:** ci si riferirà al progetto solo con "Romeo". Per la corretta scrittura è stato creato il comando \LaTeX apposito \project ;
- **Sigle:** le sigle andranno usate *esclusivamente* all'interno di tabelle o diagrammi secondo il seguente formalismo:
 - Re per indicare il *Responsabile di Progetto*;
 - Ad per indicare il *Amministratore di Progetto*;
 - Ve per indicare i *Verificatori*;
 - Pr per indicare i *Programmatori*;
 - Pj per indicare i *Progettisti*;
 - An per indicare gli *Analisti*.

3.1.2.6 Componenti grafiche

¹⁰http://it.wikipedia.org/wiki/ISO_8601

¹¹Nel caso lo si possa scrivere utilizzando solo una cifra, si impone di anteporre uno zero al numero.

¹²Vedi la nota precedente.

3.1.2.6.1 Immagini: tutte le immagini utilizzate all'interno dei documenti dovranno avere formato **PDF_G** o **PNG_G**; è consigliato utilizzare **PDF_G** vettoriale in quanto la sua risoluzione non dipende dal ridimensionamento dell'immagine.

La conversione di un'immagine in formato **PDF_G** è consentita grazie all'utilizzo del software Gimp¹³.

Le immagini devono essere accompagnate da una didascalia che inizia con la parola “Figura” con la prima lettera maiuscola, seguita dal numero della figura, dal carattere “due punti” e da una breve descrizione della figura¹⁴.

3.1.2.6.2 Tabelle: Ogni tabella deve essere utilizzata all'interno dei documenti con lo scopo di proporre informazioni in modo ordinato e coerente. Le tabelle dovranno essere accompagnate da una didascalia che inizia con la parola “Tabella” con la prima lettera maiuscola, seguita dal numero della tabella, dal carattere “due punti” e da una breve descrizione non banale della tabella¹⁵.

Nel caso in cui il valore di una cella sia zero e il dato non è significativo, esso verrà omesso.

3.1.3 Produzione

Lo scopo di quest'attività è di definire le norme riguardanti la produzione della documentazione definita nell'attività di *Pianificazione* (vedi sez. 3.1.1).

3.1.3.1 Automatizzazione dei paragrafi al fine di ridurre l'impiego di risorse nell'attività di verifica della documentazione, è necessario automatizzare la produzione di alcuni paragrafi dei documenti. Per la loro struttura modulare, sono stati individuati due paragrafi che possono essere automatizzati: *Componenti e classi* del documento **Specifica Tecnica** (vedi 3.1.2.4.6) e *Specifica componenti* del documento **Definizione di Prodotto** (vedi 3.1.2.4.7).

I *Progettisti* sono dunque tenuti ad utilizzare l'applicativo **reqMonkeys** (vedi sez.5.2.3), per memorizzare nel sistema la struttura di Romeo. Sarà possibile quindi scaricare dall'applicativo il file **.TEX** relativo al paragrafo *Componenti e Classi*, per poterlo inserire successivamente nella struttura del documento. Allo stesso modo, i *Programmatori* sono tenuti a memorizzare in **reqMonkeys** i dati riguardanti il codice mano a mano sviluppato (classi, relazioni, note). Sarà possibile quindi scaricare dall'applicativo il file **.TEX** relativo al paragrafo *Specifica componenti*, per poterlo inserire successivamente nella struttura del documento.

3.2 Processo di verifica

Il processo di verifica è un processo che consente di determinare se i risultati di un'attività soddisfano i requisiti fissati nelle attività precedenti. Per minimizzare i costi e massimizzare l'efficacia, la verifica dovrebbe essere integrata il prima possibile nel processo. Tale processo si suddivide nelle seguenti attività:

- **Pianificazione;**
- **Verifica;**

3.2.1 Pianificazione:

3.2.1.1 Tecniche di analisi: esistono due tecniche di analisi: l'*analisi statica* e l'*analisi dinamica*.

¹³<http://www.gimp.org>

¹⁴Con breve si vuole intendere che la descrizione deve stare in una riga.

¹⁵Vedi nota precedente.

3.2.1.1.1 Analisi statica: può essere di tipo walkthrough o inspection.

- **Walkthrough:** questa tecnica di analisi prevede una lettura critica del codice o del documento prodotto. Tale tecnica è molto dispendiosa in termini di risorse, visto che viene applicata all'intero documento, senza avere una precisa idea di quale sia il tipo di anomalia e di dove ricercarla. Essa è però necessaria nelle prime fasi del progetto, vista l'inesperienza da parte del gruppo nell'attuare un tipo di verifica più precisa e mirata. Dopo una prima fase di lettura ed identificazione degli errori, si procede alla discussione degli stessi, proponendo le modifiche da apportare per garantirne la correzione. Il passo finale consiste nell'applicare le modifiche proposte, redigendo un rapporto preciso che elenchi le modifiche effettuate. Una caratteristica di questo tipo di analisi è che richiede l'utilizzo di più risorse umane;
- **Inspection:** questa tecnica di analisi presuppone l'esperienza da parte del verificatore, nell'individuare gli errori e le anomalie più frequenti. A tal scopo è necessaria una *lista di controllo*, stilata in una precedente analisi di tipo walkthrough, nella quale vengono elencate le sezioni più critiche. Questo permette quindi una verifica più rapida che non necessiti della lettura dell'intero documento o codice, oltre che richiedere meno risorse umane. Dopo aver terminato l'analisi, sarà necessario stilare un rapporto di verifica che tenga traccia del lavoro svolto e delle modifiche apportate.

3.2.1.1.2 Analisi dinamica: si applica solo al prodotto software e consiste nella verifica e validazione attraverso dei **test**. Per garantire la correttezza dei test è necessario che essi siano **ripetibili**: solamente dei test che, dato un certo input, producono sempre lo stesso output, riescono a verificare la correttezza del prodotto.

Prima di effettuare questo tipo di analisi, devono essere definiti:

- **Ambiente:** consiste sia del sistema software che del sistema hardware utilizzato;
- **Specifica dei valori:** consiste nel definire sia i parametri di input necessari per la corretta esecuzione del test, sia il valore atteso per i parametri di output;
- **Procedure:** dev'essere definito il modo in cui i test devono essere effettuati, specificando un eventuale ordine e come dovranno essere interpretati i risultati.

Di seguito saranno esposti i vari tipi di test che verranno utilizzati:

- **Test di unità:** verifica ogni singola unità del prodotto software, avvalendosi di strumenti come driver_G e/o stub_G. In particolare si verifica che per ogni unità siano esplorati tutti i cammini di esecuzione possibili e che i requisiti per quella determinata unità siano soddisfatti;
- **Test di integrazione:** per i moduli che hanno soddisfatto i test di unità, si procede alla verifica di più unità combinate, con lo scopo di evidenziare eventuali errori residui. Per effettuare tali test si farà uso di classi appositamente create per simulare e verificare l'interazione;
- **Test di sistema:** consiste nella validazione del sistema, attraverso la verifica delle coperture di tutti i requisiti individuati durante l'**Analisi Incrementale (FAI)**;
- **Test di regressione:** consiste nell'eseguire nuovamente i test di unità e di integrazione per le componenti che hanno subito modifiche, così da verificare che non siano stati introdotti nuovi errori e/o anomalie;
- **Test di accettazione:** consiste nel collaudo del sistema eseguito dal committente. Se l'esito di tale collaudo è positivo si può procedere al rilascio ufficiale del prodotto.

3.2.2 Verifica:

quest'attività consiste nella verifica dei prodotti delle varie attività, in maniera da concludere se sono conformi o meno ai requisiti fissati precedentemente per quei prodotti.

3.2.2.1 Analisi statica del codice: per la verifica statica in Jenkins sono stati integrati i seguenti strumenti:

- **Compilatori:** il compilatore è il primo strumento di analisi, in quanto segnala in modo inequivocabile errori e warning. Dato che il software dovrà essere multiplatforma, i sorgenti del progetto dovranno essere compilati con i seguenti compilatori:
 - **GCC**¹⁶: compilatore per ambiente Linux_G;
 - **Clang**¹⁷: compilatore per ambiente Mac OS_G;
 - **MinGW**¹⁸: compilatore per ambiente Windows_G.
- **CppCheck**¹⁹: si tratta di un analizzatore statico di codice per il C++_G, che permette la rilevazione di alcuni errori;
- **CCCC**²⁰ *C and C++ Code Counter*: Misura metriche riguardanti codice sorgente C++_G. Tra le più rilevanti ci sono: complessità ciclomatica, linee di codice per linee di commento, metodi per classe ed accoppiamento tra gli oggetti.

3.2.2.2 Analisi dinamica del codice: l'analisi dinamica verrà effettuata con:

- **Valgrind**²¹: tool_G che permette di rilevare memory leak_G, altri problemi di memoria comuni in C++_G ed effettuare il profiling_G del software. Compatibile solo con sistemi Unix;
- **Very Sleepy**: tool per il profiling_G del software in ambiente Windows_G;
- **Dr. Memory**²²: tool per il controllo della memoria in ambiente Windows_G;

3.2.2.3 Test d'unità: per i test di unità sul codice si utilizzerà il framework_G Qt Test²³.

Per maggiori informazioni si rimanda al *Piano di Qualifica v3.0.0*.

3.2.2.4 Verifica dei diagrammi: al *Verificatore* è richiesta la verifica dei seguenti diagrammi UML_G:

- **Diagrammi di flusso:** la verifica dei diagrammi di flusso deve avvenire manualmente controllando che essi aderiscano allo standard UML_G 2.0 e rappresentino in modo corretto la procedura in questione;
- **Diagrammi dei casi d'uso:** la verifica dei diagrammi dei casi d'uso deve avvenire manualmente, controllando che essi rispettino lo standard UML_G 2.0. In particolare, deve essere verificato il corretto utilizzo di inclusioni ed estensioni; inoltre, si deve assicurare che ci sia la massima corrispondenza tra il diagramma e la descrizione testuale.

¹⁶<http://gcc.gnu.org>

¹⁷<http://clang.llvm.org>

¹⁸<http://www.mingw.org>

¹⁹<http://cppcheck.sourceforge.net>

²⁰<http://cccc.sourceforge.net>

²¹<http://valgrind.org>

²²<http://www.drmemory.org>

²³<http://qt-project.org/doc/qt-5.0/qttestlib/qtest-overview.html>

3.2.2.5 Verifica della documentazione: la verifica della documentazione viene eseguita ogni qualvolta un documento abbia subito una modifica e debba essere approvato. È compito del *Responsabile di Progetto* affidare ai *Verificatori* la verifica di un documento. Per eseguire una corretta verifica di un documento va seguita la seguente prassi:

1. **Controllo tipografico:** tramite l'utilizzo di *TeXmaker* e di *AspellG* verranno trovati errori tipografici presenti nel documento;
2. **Controllo lessicale:** il *Verificatore* dovrà controllare che il documento non contenga degli errori lessicali attraverso un'attenta lettura delle parti da verificare: se non è ancora presente una lista di controllo, si effettuerà un'analisi di tipo *walkthrough*, altrimenti si procederà all'analisi *inspection*;
3. **Controllo di glossario:** il *Verificatore* dovrà controllare che ogni termine presente nel glossario sia evidenziato correttamente all'interno del documento;
4. **Controllo di contenuto:** il *Verificatore* dovrà controllare che il documento contenga tutti gli argomenti da trattare, che non manchi nulla, e che sia impaginato con una struttura adeguata;
5. **Rispetto delle norme di progetto:** il *Verificatore* dovrà controllare che il documento segua le norme di progetto stabilite;
6. **Stesura lista di controllo:** si dovrà stilare una lista degli errori più frequenti, in modo da facilitare le successive attività di verifica dei documenti;
7. **Calcolo dell'indice di Gulpease:** su ogni documento redatto il *Verificatore* dovrà calcolare l'indice di Gulpease. Nel caso in cui l'indice non rispettasse il range di valori specificato nel *Piano di Qualifica v3.0.0*, sarà necessario eseguire un *walkthrough* del documento alla ricerca delle frasi troppo lunghe o complesse;
8. **Segnalazione degli errori riscontrati:** una volta terminata la verifica il *Verificatore* dovrà creare i ticket opportuni per segnalare gli errori riscontrati.

3.2.2.6 Comunicazione e risoluzione di anomalie e discrepanze Nel corso delle attività di verifica, si possono riscontrare delle anomalie e/o discrepanze.

- **Anomalia:** lo stato del prodotto non corrisponde a quelle che sono le sue aspettative; questo causa un totale malfunzionamento dell'intero sistema quindi, la gestione di tale problema assume una priorità alta nella gestione dei problemi;
- **Discrepanza:** si intende un allontanamento tra lo stato del prodotto attuale con le aspettative prefissate per quel prodotto; questo tipo di problema non comporta un malfunzionamento totale del sistema, ma lo rende inesatto al soddisfacimento dei requisiti specificati. Una discrepanza può essere vista come un'anomalia con priorità medio/bassa a seconda della gravità della stessa.

Per la gestione di questi problemi verrà utilizzato il sistema di ticketing offerto da *GitHubG*²⁴. Ogni qualvolta il verificatore riscontri un'anomalia, dovrà aprire un ticket apposito, riportante la descrizione dettagliata del problema.

Il *Responsabile di Progetto*, prima di approvare tale ticket, dovrà verificare che l'anomalia sussista e che questa non sia già compresa in altri ticket precedentemente approvati. Si riserva al Responsabile inoltre, la facoltà di non approvare tale ticket, se il costo per effettuare l'operazione di modifica, non fosse compensabile con il beneficio derivante dalla stessa. Le anomalie e le discrepanze potranno riguardare:

- Mancato rispetto con quanto indicato nelle *Norme di Progetto v3.0.0*;
- Mancato rispetto dei valori di accettazione da parte di una delle misurazioni (vedi ??);

²⁴Per una descrizione dettagliata fare riferimento a documento *Norme di Progetto v3.0.0*

- Allontanamento dai requisiti specificati durante l'analisi dei requisiti.

Si rimanda alla consultazione delle *Norme di Progetto v3.0.0* per le modalità di apertura di un ticket.

3.3 Processo di garanzia della qualità

Il processo di garanzia della qualità, è un processo che consente di garantire che il prodotto software ed i processi del ciclo di vita dello stesso, siano conformi ai loro requisiti e che aderiscano ai piani prestabiliti. Tale processo si suddivide nelle seguenti attività:

- **Qualità del prodotto;**
- **Qualità del processo;**

3.3.1 Qualità del prodotto

Per controllare la qualità del prodotto dovranno essere svolte le seguenti attività:

- **Quality assurance:** si verifica che il prodotto soddisfi gli obiettivi di qualità prefissati;
- **Verifica:** attività svolta durante l'intera durata del progetto, ha come scopo quello di controllare che il prodotto corretto e congruo alle aspettative;
- **Validazione:** viene verificato che il prodotto risponda ai requisiti specificati.

3.3.2 Qualità del processo

La qualità di un prodotto dipende dalla qualità del processo da cui proviene. Per garantire la qualità dei processi è necessario che i processi siano dettagliatamente pianificati e che ci sia un controllo continuo su di essi. Si è quindi deciso di aderire allo standard ISO_G/IEC_G15504, conosciuto anche con l'acronimo **SPICE** (Software Process Capability dEtermination)²⁵. La qualità dei processi verrà garantita inoltre, adottando il ciclo di Deming_G(PDCA). Il ciclo di Deming è un modello incrementale studiato per il miglioramento continuo della qualità dei processi. Si fonda sulla suddivisione in quattro fasi:

- **Plan:** si stabiliscono gli obiettivi e i processi necessari per raggiungere le aspettative prefissate;
- **Do:** si implementa il piano, eseguendo il processo, realizzando il prodotto;
- **Check:** si studiano i risultati ottenuti comparandoli con i risultati attesi;
- **Act:** si procede alla correzione dei processi, determinando la causa della discrepanza tra i risultati ottenuti e quelli attesi.



Figura 5: Il ciclo PDCA

²⁵Per una descrizione dettagliata dello standard, consultare l'appendice ??

4 Processi organizzativi

4.1 Processo di gestione del progetto

Il processo di gestione del progetto contiene tutte le attività ed i compiti che consentono la gestione del processo di sviluppo (vedi sezione 2.1). Il *Responsabile di Progetto* avrà il compito di pianificare le varie attività, gestire le risorse ed Analizzare e prevenire i rischi.

Il processo di gestione del progetto è composto dalle seguenti attività:

- **Pianificazione;**
- **Esecuzione e controllo;**

4.1.1 Pianificazione

Lo scopo di quest'attività è di preparare un piano di esecuzione del processo. Il *Responsabile di Progetto* ha il compito di redigere tale piano e di produrre la relativa documentazione (vedi paragrafi relativi al *Piano di Progetto* nelle sezioni 3.1.1 e 3.1.2).

4.1.1.1 Pianificare attività: per pianificare le varie attività da svolgere, il *Responsabile di Progetto* dovrà utilizzare gli strumenti identificati nel paragrafo dedicato ai diagrammi di Gantt_G nella sezione 4.2.1. Tali diagrammi dovranno essere inclusi nella documentazione.

4.1.1.2 Gestione delle risorse: per gestire le varie risorse disponibili durante lo svolgimento del progetto, il *Responsabile di Progetto* dovrà creare dei diagrammi delle risorse, per pianificare la quantità di ore che ogni risorsa dovrà dedicare a ciascuna attività. Per disegnare i diagrammi di attività, si dovranno utilizzare gli strumenti identificati nel paragrafo dedicato ai diagrammi di Gantt_G nella sezione 4.2.1. Tali diagrammi dovranno essere inclusi nella documentazione. Inoltre, per assegnare i compiti alle risorse disponibili, il *Responsabile di Progetto* utilizzerà il sistema di ticketing offerto da GitHub_G²⁶.

4.1.1.2.1 Rotazione dei ruoli: durante lo svolgimento del progetto, è necessario che tutti i componenti del gruppo ricoprano tutti i vari ruoli definiti nel *Piano di Progetto v3.0.0*. Nel precedente documento verranno inoltre specificate le assegnazioni dei ruoli alle risorse, ripartite temporalmente e il quantitativo di ore che ciascuna risorsa dovrà svolgere in veste di un determinato ruolo.

Per pianificare l'impiego delle risorse in tale senso, il *Responsabile di Progetto* si ispirerà alle seguenti regole:

- Ogni risorsa dovrà ricoprire tutti i ruoli;
- Il carico di ore individuali dovrà essere equo.

Riportiamo di seguito le direttive che regolano l'assegnazione e la modalità di rotazione dei ruoli.

I ruoli verranno assegnati alle risorse in modo tale che cambino al termine di una macro-fase (vedi *Piano di Progetto v3.0.0*). Dato che i ruoli da ricoprire sono in quantità minore rispetto alle risorse, sarà necessario che una o più risorse ricoprano più ruoli durante la stessa macro-fase. Per regolamentare questa ulteriore rotazione, le risorse ricopriranno un singolo ruolo per settimana, ruotando tra i ruoli assegnati nella specifica macro-fase.

Ogni componente del gruppo potrà consultare, in qualsiasi momento, i diagrammi di Gantt_G che descrivono la gestione delle risorse e dei ruoli, in maniera tale che ognuno potrà sempre essere consapevole del ruolo ricoperto dagli altri componenti.

²⁶Per maggiori informazioni consultare la sezione 5.4.3.

4.1.1.3 Analisi e prevenzione dei rischi: durante l'intero periodo di svolgimento del progetto, il *Responsabile di Progetto* dovrà analizzare i rischi che possono incombere. Inoltre dovrà cercare di prevenirli e trovare delle contromisure ad essi. Tale analisi dovrà essere inclusa nella relativa documentazione.

4.1.2 Esecuzione e controllo

4.1.2.1 Incontri: sarà il *Responsabile di Progetto* a fissare gli incontri, sia interni che esterni.

Per ogni incontro dovrà essere specificata data, ora, luogo, ordine del giorno e motivi della riunione; tali informazioni dovranno essere rese disponibili con almeno quattro giorni di anticipo.

Successivamente alla decisione di una nuova data il *Responsabile di Progetto* provvederà alla creazione di un evento sul calendario condiviso su Google Calendar_G.

4.1.2.1.1 Incontri esterni: sarà il *Responsabile di Progetto* a fissare gli incontri con i proponenti e/o committenti utilizzando la casella di posta creata appositamente²⁷. Il *Responsabile di Progetto* prima di prendere alcun accordo con le parti esterne dovrà contattare i vari componenti del gruppo, per sentire se almeno cinque membri concordano. In caso positivo il *Responsabile di Progetto* provvederà a contattare i proponenti e/o committenti per fissare la data dell'incontro. Sarà compito del *Responsabile di Progetto* redigere il verbale dell'incontro avvenuto.

4.1.2.1.2 Incontri interni: sarà il *Responsabile di Progetto* a fissare gli incontri interni, contattando tutti i membri del gruppo. Gli incontri interni dovranno avere una frequenza almeno quindicinale. Ogni componente del gruppo è tenuto a leggere regolarmente la posta elettronica personale e rispondere ad eventuali richieste di un incontro interno. Nel caso in cui i membri disponibili a partecipare alla riunione in una certa data siano meno di quattro, questa verrà posticipata o anticipata, in modo che siano presenti un numero adeguato di persone. Inoltre, è possibile e auspicabile che siano necessarie riunioni tra specifici membri del gruppo, ad esempio: in fase di analisi può essere utile che solo gli *Analisti* si incontrino tra di loro, senza il resto del gruppo. I restanti componenti del gruppo saranno comunque informati sui contenuti e le decisioni prese tramite invio di una e-mail alla mailing list_G o alla pubblicazione di un verbale sul repository_G nel caso siano state prese decisioni importanti.

4.1.2.1.3 Richieste di incontri: qualora ve ne sia la necessità, ogni membro del gruppo potrà richiedere un incontro sia interno che esterno, contattando personalmente il *Responsabile di Progetto* ed esponendo i motivi della richiesta. Il *Responsabile di Progetto* avrà potere di scegliere se accettare o rifiutare la richiesta.

4.1.2.2 Creazione milestone: il *Responsabile di Progetto* dovrà provvedere alla creazione di una milestone_G per la prossima revisione a cui il gruppo *Seven Monkeys* ha intenzione di partecipare. La gestione di una milestone_G prevede l'apertura di una serie di ticket, atti a pianificare i compiti da svolgere entro la chiusura della stessa (vedi figura 6). Per la procedura di apertura di una milestone si veda il paragrafo *Creazione milestone* in sezione ??.

4.1.2.3 Creazione ticket: i ticket vengono creati da:

- **Responsabile di Progetto:** crea la maggioranza dei ticket;

²⁷Si veda il paragrafo *Strumenti di comunicazione* della sezione 4.2.1

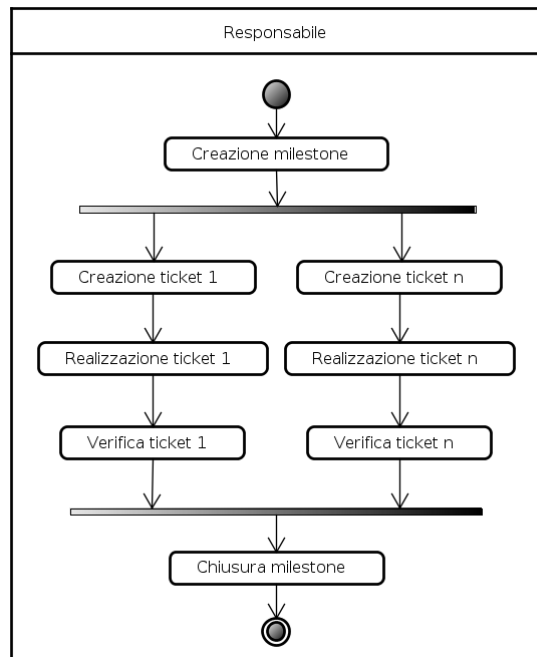


Figura 6: Modello per la gestione di una milestone

- **Verificatore:** crea i ticket per segnalare le imprecisioni o errori trovati durante la verifica.

Ogni ticket dovrà essere assegnato ad un unico membro del team *Seven Monkeys* e dovrà avere le seguenti caratteristiche²⁸:

- **Destinatario:** a chi è rivolto l'attività indicata dal ticket;
- **Milestone:** la milestone_G a cui è associato il ticket;
- **File:** dovrà essere indicato il file oggetto dell'attività;
- **Label:** ogni ticket avrà una o più label associate; le label possibili sono le seguenti:
 - **Modifica:** ticket generalmente creato dal *Verificatore* per segnalare gli errori trovati;
 - **Verifica:** ticket creato dal *Responsabile di Progetto* per assegnare la verifica a uno dei *Verificatori*;
 - **Richiesta approvazione:** ticket creato da un *Verificatore* per richiedere l'approvazione di un documento;
 - **Creazione:** ticket creato dal *Responsabile di Progetto* per assegnare un compito a un generico membro del gruppo;
 - **Priorità alta:** label assegnata ad un ticket creato per la gestione di anomalie;
 - **Priorità media:** label assegnata ad un ticket creato per la gestione di discrepanze di media gravità;
 - **Priorità bassa:** label assegnata ad un ticket creato per la gestione di discrepanze non gravi.

Si vedano le figure 7 e 8, per i diagrammi di attività della gestione dei ticket. Per la procedura di apertura di un ticket, si veda il paragrafo *Creazione ticket* nella sezione ??.

²⁸In figura 3 è mostrata l'interfaccia per la creazione di ticket.

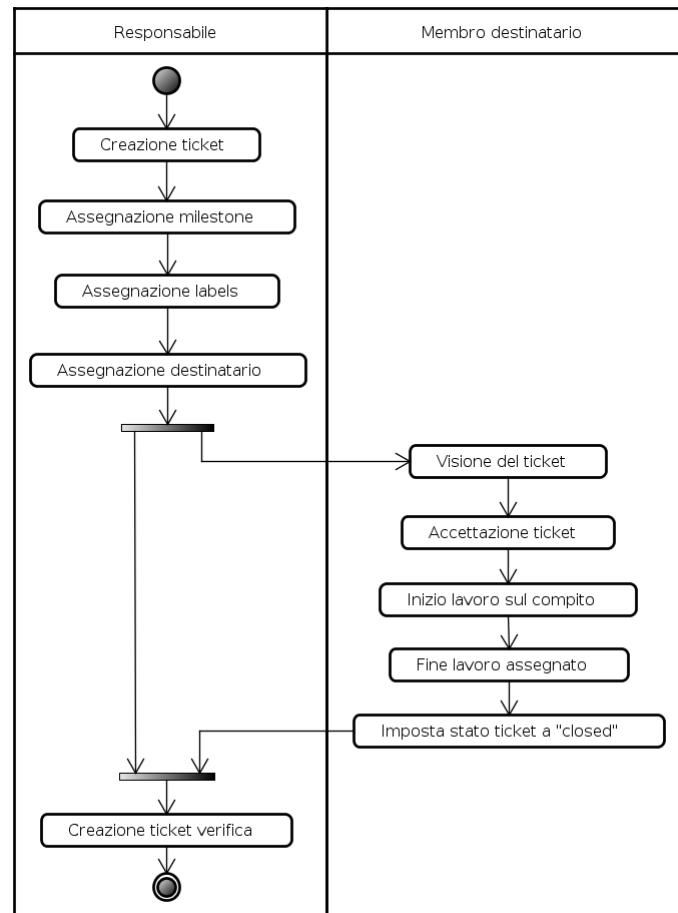


Figura 7: Modello di ticket per la creazione di un compito generico

4.1.2.4 Esecuzione dei compiti: ogni membro del gruppo dovrà visionare i ticket a lui assegnati e, successivamente all'esecuzione del compito, dovrà cambiare lo stato del ticket in "closed". Quando il destinatario del ticket ne prenderà visione, dovrà confermarlo con un commento nella pagina web del ticket. Se il *Responsabile di Progetto* si rende conto che un ticket non è stato eseguito come atteso, "riaprirà" il ticket.

4.1.2.5 Chiusura milestone: nel momento in cui tutti i ticket riferiti ad una milestone_G sono stati completati e chiusi, sarà cura del *Responsabile di Progetto* verificarne l'effettiva chiusura e di conseguenza chiudere la milestone_G. Si veda il paragrafo *Chiusura milestone* della sezione ?? per la relativa procedura.

4.2 Processo di organizzazione dell'infrastruttura

Il processo di organizzazione dell'infrastruttura consente di fornire servizi ed infrastrutture operative in grado di supportare le esigenze dei processi che il gruppo *Seven Monkeys* intende implementare durante tutto il ciclo di vita del software. Questo processo definisce, fornisce e mantiene i mezzi e le tecnologie chiave di cui ha bisogno il team per lo svolgimento delle proprie attività.

Il processo di organizzazione dell'infrastruttura è composto dalle seguenti attività:

- **Definizione dell'infrastruttura;**

4.2.1 Definizione dell'infrastruttura

Lo scopo di quest'attività è di individuare quali infrastrutture sono necessarie per lo svolgimento dei vari processi. Nei seguenti paragrafi verranno definiti gli strumenti, individuati

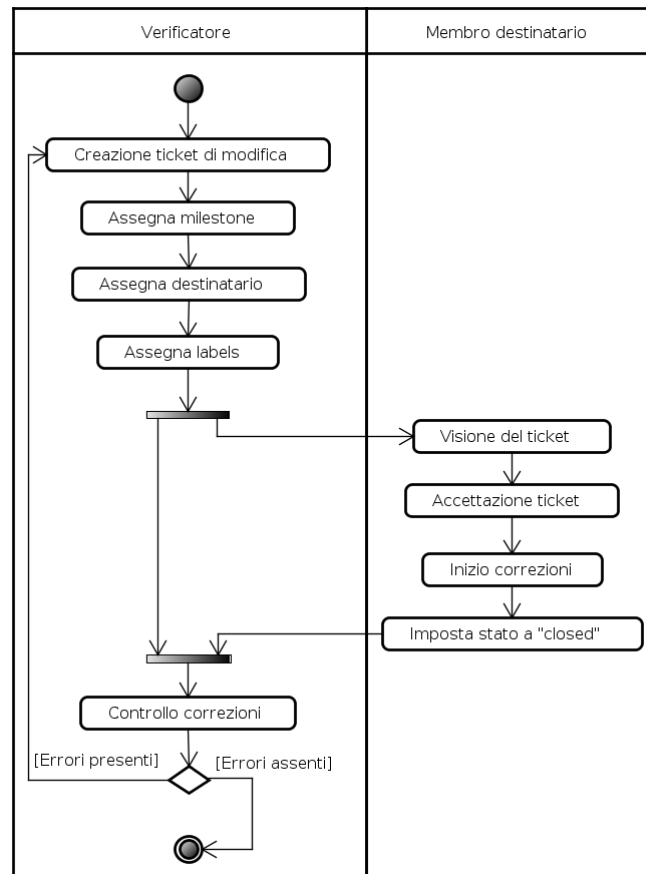


Figura 8: Modello di ticket per la creazione di un ticket di verifica

dagli *Amministratori*, di cui il team necessita per lo svolgimento di un progetto. Ogni strumento dovrà essere documentato per quanto riguarda l'installazione e l'esecuzione delle procedure ricorrenti (vedi 5).

4.2.1.1 Ambiente di sviluppo: il sistema operativo utilizzato per lo sviluppo del progetto è a discrezione di ogni singolo componente del gruppo. I vari membri del gruppo utilizzeranno i seguenti sistemi operativi:

- Ubuntu 12.04 64bit;
- Mac OS_G 10.9 64bit;
- Windows_G 8 64bit.

4.2.1.2 Controllo di versione: è necessario usufruire di un controllo di versione per lo sviluppo della documentazione e del codice. Per questo verranno creati due repository_G distinti. Essi saranno privati e quindi accessibili solo ai membri del team.

I repository_G saranno ospitati dal servizio GitHub_G, il quale utilizza il sistema di versionamento Git_G. Per quanto riguarda le procedure di interazione e di installazione, vedere il paragrafo 5.1.1.

4.2.1.2.1 Repository della documentazione: all'interno di questo repository_G verranno memorizzati tutti i file necessari alla generazione dei documenti. Di seguito verrà descritta sinteticamente l'organizzazione interna del filesystem.

- **Consegne:** contiene tutti i documenti in formato PDF_G consegnati nelle varie revisioni. I file sono raggruppati in sotto cartelle denominate con l'acronimo della revisione a cui fanno riferimento. La struttura delle sottocartelle sarà la seguente:

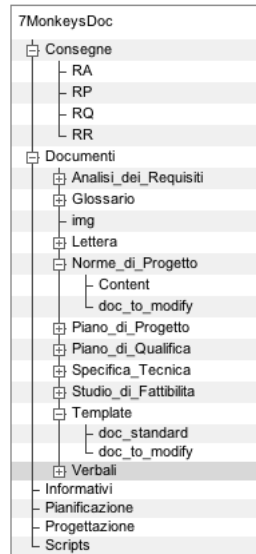


Figura 9: Organizzazione file system del repository dei documenti.

- **RR:** Revisione dei Requisiti;
 - **RP:** Revisione di Progettazione;
 - **RQ:** Revisione di Qualifica;
 - **RA:** Revisione di Accettazione.
- **Documenti:** è la cartella contenente i sorgenti di tutti i documenti, sia in stato di redazione, sia completati. Il suo contenuto è organizzato nel seguente modo:
 - **Nome_del_Documento:** ogni documento in fase di redazione dovrà avere la propria cartella, denominata con lo stesso nome del documento (vedi 9). Se il documento necessita di immagini, esse dovranno essere salvate in una sottocartella denominata “Immagini”;
 - **Template:** contiene i template \LaTeX comuni a tutti i documenti necessari per la loro compilazione. In particolare, in essi vengono definiti gli stili delle pagine, il frontespizio e i vari comandi creati per la stesura;
 - **Verbali:** contiene i verbali redatti dal gruppo durante lo svolgimento del progetto. Per ogni documento verrà creata una sottocartella apposita, contenente i file necessari.
 - **Informativi:** contiene dei file informativi; per esempio è presente una procedura per la creazione di un nuovo documento e un file contenente la lista di tutti i comandi \LaTeX utilizzati nei file di compilazione;
 - **Pianificazione:** contiene tutti i grafici Gantt_G utilizzati nella pianificazione delle attività e nella gestione delle risorse;
 - **Progettazione:** contiene tutti i diagrammi UML_G utilizzati durante la progettazione del software;
 - **Scripts:** contiene gli script utilizzati per automatizzare alcune procedure; per esempio la compilazione dei documenti per una consegna.

4.2.1.2.2 Repository del codice: all’interno di questo repository_G verranno memorizzato il codice sorgente del software e la relativa documentazione. Di seguito verrà descritta sinteticamente l’organizzazione interna del filesystem.

- **Romeo:** è la cartella principale, contenente i sorgenti dell’applicativo Romeo. Il suo contenuto è organizzato come segue:

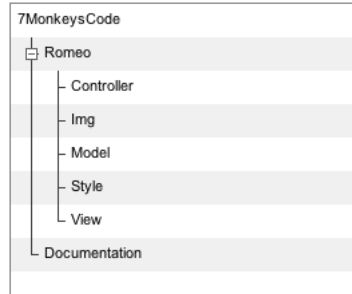


Figura 10: Organizzazione file system del repository del codice.

- **Controller:** contiene il codice relativo al componente *Controller* dell'architettura software;
- **Img:** contiene le immagini che faranno parte della GUI_G del software;
- **Model:** contiene il codice relativo al componente *Model* dell'architettura software;
- **Style:** contiene i fogli di stile relativi alle finestre dell'applicativo;
- **View:** contiene il codice relativo al componente *View* dell'architettura software.
- **Documentation:** contiene i file di documentazione, generati dalla compilazione del codice sorgente.

4.2.1.3 Stesura documentazione: è necessario utilizzare un sistema di stesura dei documenti che venga incontro alle esigenze tipografiche, di personalizzazione e che inoltre permetta il controllo di versione dei file. Conseguentemente, potrà essere individuato un editor testuale adeguato alle esigenze di sviluppo. Verrà quindi utilizzato il linguaggio di markup_G \LaTeX (vedi sezione 5.2.2) e l'editor *Tex-Maker* (vedi sezione 5.2.4).

4.2.1.4 Diagrammi UML: è necessario individuare gli strumenti adatti per la produzione dei grafici UML_G che saranno inseriti nella documentazione. Tali diagrammi dovranno essere conformi allo standard UML_G 2.0. Verranno quindi utilizzati: *Astah_G - Professional Edition* per la produzione dei diagrammi dei casi d'uso e di attività (vedi sezione 5.2.1) e *Visual Paradigm* per i diagrammi di package e di classe (vedi sezione 5.2.5).

4.2.1.5 Diagrammi di Gantt: è necessario individuare gli strumenti adatti per pianificare la gestione di progetto e gestire le risorse umane. In particolare, sarà necessario produrre dei diagrammi di Gantt_G che andranno inseriti nella documentazione. Verrà quindi utilizzato *GanttProject_G* (vedi sezione 5.4.2).

4.2.1.6 Codifica: è necessario individuare gli strumenti adatti per la codifica del codice. Gli strumenti possono comprendere:

- **Librerie esterne:** eventualmente individuate durante la progettazione architetturale (vedi 2.1.2);
- **Framework:** costituiscono eventualmente la base su cui sviluppare il prodotto, da individuare in base alle esigenze di progettazione;
- **Compilatori:** da individuare in base ai linguaggi utilizzati;
- **IDE:** eventualmente utilizzati per facilitare la codifica, da individuare in base ai linguaggi utilizzati;
- **Documentazione del codice:** strumenti in grado di generare una documentazione efficace del codice sviluppato, da individuare in base al linguaggio utilizzato;

In base alle esigenze di cui sopra ed al tipo di progetto da sviluppare, sono state individuati i seguenti strumenti:

- ITK_G, VTK_G come librerie esterne di supporto;
- Qt_G come framework_G su cui basare lo sviluppo dell'applicativo;
- G++ e CMake come compilatori;
- Qt Creator e Qt Designer come IDE_G per lo sviluppo;
- Doxygen per la generazione della documentazione del codice;

Vista la varietà di applicativi individuati per lo sviluppo, gli *Amministratori* hanno ritenuto opportuno configurare una macchina virtuale, contenente tutti gli strumenti necessari, in maniera da minimizzare lo sforzo di configurazione dei singoli membri del team e superare i problemi legati allo sviluppo su diversi sistemi operativi. Tale macchina virtuale, basata su Ubuntu 12.04 LTS, è stata costruita utilizzando l'applicativo VMware²⁹. Si faccia riferimento alla sezione 5.1.2 per le procedure di installazione e utilizzo.

4.2.1.7 Strumenti di coordinamento: è necessario individuare gli eventuali strumenti a supporto del coordinamento tra i membri del team. Tra questi, sono compresi gli applicativi per la condivisione di file non soggetti a versionamento e gli applicativi per la gestione degli impegni del team.

Alla luce delle esigenze individuate, verranno quindi utilizzati i seguenti strumenti:

- **Dropbox:** in questo servizio cloud verranno messi tutti i file che non sono soggetti ad un controllo di versione. Verrà principalmente utilizzato dai membri del gruppo per scambiarsi libri di testo, guide, ecc. . . . Si faccia riferimento alla sezione 5.4.1 per i dettagli e le procedure di supporto.
- **Google Drive:** in questo servizio cloud andranno inseriti tutti i documenti che:
 - Non necessitano di alcun controllo di versione;
 - Necessitano di una forte interattività tra i vari membri del gruppo;
 - Sono accessibili direttamente da browser.

Principalmente Google Drive_G viene utilizzato per lavorare su dei file condivisi su Google Docs_G. Si faccia riferimento alla sezione 5.4.5 per i dettagli e le procedure di supporto.

- **Google Calendar:** viene utilizzato dai membri del gruppo per la gestione delle risorse umane. Verrà creato un calendario condiviso tra i vari membri del gruppo, in modo da notificare:
 - In quali date un certo membro non è disponibile;
 - In quali date c'è una riunione o un evento rilevante ai fini del gruppo.

²⁹<http://www.vmware.com/>

Grazie alla gestione delle notifiche di Google Calendar_G, è possibile far in modo che 24 ore prima di un evento rilevante venga inviata una email a tutti i membri del gruppo come promemoria. Si faccia riferimento alla sezione 5.4.4 per i dettagli e le procedure di supporto.

4.2.1.8 Strumenti di comunicazione: è necessario individuare gli strumenti di comunicazione adatti a soddisfare le esigenze del team. Tra questi, sono compresi gli strumenti a supporto delle comunicazioni formali ed informali.

I seguenti paragrafi, esplicheranno le scelte prese alla luce delle esigenze individuate.

4.2.1.8.1 Comunicazioni formali: le comunicazioni formali possono essere esterne o interne.

Per le comunicazioni verso l'esterno è stato creato un indirizzo di posta elettronica apposito: seven.monkeys.swe@gmail.com. Tale indirizzo dovrà essere l'unico servizio utilizzabile per le comunicazioni verso l'esterno. Sarà solo il *Responsabile di Progetto* ad utilizzare l'indirizzo di posta per conto del gruppo *Seven Monkeys* intrattenendo le corrispondenze con i proponenti ed i committenti. Eventualmente, il *Responsabile di Progetto* provvederà ad inoltrare le conversazioni a tutti i membri del gruppo tramite la mailing list_G, ma solo se ritiene che sia necessario.

Le comunicazioni interne verranno eseguite tramite la mailing list_G: seven.monkeys.swe@gmail.com. Quando un membro del gruppo vuole inviare un'email a tutti i componenti, deve inviare il messaggio dalla sua email personale verso l'indirizzo seven.monkeys.swe@gmail.com. Un inoltro automatico provvederà a trasmettere l'email agli indirizzi personali dei componenti del gruppo presenti nella mailing list_G, tranne che al membro che ha inviato il messaggio. In questo modo tutti i componenti saranno sempre al corrente di tutti gli incontri e impegni del gruppo.

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

- **Mittente:** il mittente della email potrà cambiare a seconda del tipo di comunicazione svolta:
 - **Esterna:** l'unico indirizzo utilizzabile per comunicare verso l'esterno dovrà essere *necessariamente* l'indirizzo seven.monkeys.swe@gmail.com, il quale sarà usato esclusivamente dal *Responsabile di Progetto*;
 - **Interna:** in questo caso andrà messo l'indirizzo personale di chi scrive.
- **Destinatario:** il destinatario della e-mail cambierà a seconda che si tratti di una comunicazione interna o esterna:
 - **Esterna:** l'indirizzo del destinatario potrà variare a seconda che si voglia comunicare con il Prof. Tullio Vardanega, il Prof. Riccardo Cardin o con i proponenti del progetto;
 - **Interna:** l'unico indirizzo utilizzabile è seven.monkeys.swe@gmail.com.

Sono ammesse alcune eccezioni:

- **Proposta all'Amministratore di Progetto:** nel caso in cui un membro del gruppo voglia contattare l'*Amministratore di Progetto* per richiedere cambiamenti alle norme, il membro dovrà contattarlo al suo indirizzo di posta personale³⁰;

³⁰Sarà possibile trovare i vari recapiti e-mail personali nel documento "contatti" condiviso su Google Drive.

- **Proposta al *Responsabile di Progetto*:** nel caso in cui un membro del gruppo voglia richiedere una riunione, dovrà contattarlo al suo indirizzo di posta personale³¹;
- **Comunicazione ristretta tra alcuni membri del team:** in alcuni casi i membri del team potrebbero avere la necessità di comunicare tra di loro e utilizzeranno i loro indirizzi personali³².
- **Oggetto:** l'oggetto deve essere chiaro, esaustivo e possibilmente univoco, in modo da riconoscerlo da quelli precedenti. Nel caso si debba inviare un messaggio alla mailing list_G, vi è l'obbligo di aggiungere “**CI:**” all’inizio dell’oggetto.
- **Corpo:** il corpo di un messaggio dovrà avere tutti gli elementi e le informazioni che permettano a tutti i destinatari di capire correttamente l'argomento trattato. Se alcune parti del messaggio si riferiscono a particolari membri del gruppo o a certi ruoli di progetto si dovrà usare la seguente sintassi: “**@Cognome Nome**” o “**@Nome Ruolo**” per riferirsi ad essi. Nel caso in cui il corpo del messaggio abbia più di trenta righe, è preferibile scrivere un corpo riassuntivo ed allegare un file PDF_G che scenda più nel dettaglio. Alla fine del corpo il mittente dovrà sempre firmarsi col suo cognome, nome e ruolo.
- **Allegati:** viene consentito di allegare dei file al messaggio, preferibilmente in formato PDF_G, i quali non dovranno superare i 20MB. Essi potranno essere utilizzati ad esempio per allegare il verbale di un incontro.

4.2.1.8.3 Comunicazioni informali: al fine di facilitare le comunicazioni tra i vari membri del gruppo, viene utilizzato Google Hangout_G come servizio di messaggistica istantanea e per le video chiamate. È necessario redigere un verbale, nel caso in cui siano state prese decisioni o siano emersi dettagli inerenti allo sviluppo del progetto (si veda il relativo paragrafo nella sezione 3.1.1).

4.2.1.9 Strumenti per il tracciamento: è necessario individuare gli strumenti per il tracciamento dei requisiti, delle fonti, delle componenti, dei test e dei bug, adatti a soddisfare le esigenze del team. Tali strumenti devono automatizzare il più possibile il lavoro di tracciamento e devono consentire di costruire un database persistente e facilmente accessibile ai componenti del gruppo. Sarà cura degli *Amministratori* analizzare le esigenze del team e di prendere le dovute decisioni.

Alla luce delle precedenti considerazioni, si è deciso di sviluppare un applicativo web per il tracciamento dei requisiti, delle fonti, delle componenti e dei test. Tale applicativo è stato denominato **ReqMonkeys** (si veda la sezione 5.2.3 per i dettagli e le procedure di utilizzo). Per quanto riguarda il tracciamento dei bug, si è deciso di utilizzare l'applicativo web **Mantis**³³. Si veda la sezione 5.3.2 per i dettagli e le procedure di utilizzo.

4.2.1.10 Strumenti per la verifica: è necessario individuare gli eventuali strumenti a supporto della verifica, adatti a soddisfare le esigenze del team. La verifica necessiterà di strumenti diversi, in base al tipo di processo che si sta prendendo in considerazione. Sarà cura degli *Amministratori* individuare ed implementare le migliori soluzioni. Per la verifica ortografica dei documenti scritti in L^AT_EX, verrà utilizzato l'applicativo **Aspell**_G (si veda la sezione 5.3.1 per i dettagli implementativi).

4.2.1.11 Strumenti per l'automatizzazione: è necessario individuare gli eventuali strumenti di automatizzazione, in grado di svolgere dei compiti altrimenti dispendiosi

³¹Vedi nota precedente.

³²Vedi due note precedenti.

³³<http://www.mantisbt.org>

in termini di tempo e la cui esecuzione da parte di una persona, possa introdurre errori. Sarà cura degli *Amministratori* identificare le esigenze del team e di capire quali compiti possono essere automatizzati.

Per questo, sono stati sviluppati degli script a supporto della compilazione dei documenti ed a supporto della verifica (si veda la sezione 5.4.6 per i dettagli implementativi).

5 Strumenti

In questa sezione verranno presentati tutti gli strumenti che il gruppo *Seven Monkeys* utilizzerà per svolgere le varie attività che compongono i processi. Per ogni strumento verranno esplicate le principali procedure di interazione. Le procedure di installazione si riferiscono ad **Ubuntu 12.04 LTS**.

5.1 Strumenti a supporto del processo di sviluppo

5.1.1 Git

Git è un sistema per il controllo di versione di un filesystem. Verrà adottato per ospitare i file del progetto sottoposti a controllo di versione.

Versione utilizzata: **1.8.3.2**

5.1.1.1 Installazione: per l'installazione dell'applicativo, seguire la seguente procedura:

- Aprire il terminale;
- Digitare `sudo apt-get install git-core`;
- Seguire la procedura di installazione fino al termine.

5.1.1.2 Best Practise di interazione: per evitare conflitti nel repository_G, è indispensabile eseguire le seguenti operazioni di sincronizzazione all'inizio e alla fine di ogni sessione di lavoro (fig 11):

1. **Pull:** prima di iniziare la sessione di lavoro, per ottenere i file aggiornati;
2. **Commit:** al termine di una sessione di lavoro, per commentare le modifiche effettuate;
3. **Pull:** subito prima di eseguire l'operazione di push;
4. **Push:** subito dopo l'operazione di pull, per caricare le modifiche nel repository_G remoto;
5. **Merge:** se durante l'operazione di push sorgono dei conflitti a seguito di modifiche effettuate da un altro membro.

I commenti aggiunti al commit dopo la modifica di un file devono essere chiari e devono specificare le modifiche effettuate. Per garantire una buona leggibilità dei commenti, i commit devono essere effettuati per ogni file modificato o gruppo di files dello stesso contesto. Inoltre, per evitare conflitti, è sconsigliato lavorare contemporaneamente sugli stessi file.

5.1.2 VMware

Software che permette l'emulazione di un determinato sistema operativo. Gli *Amministratori* hanno creato un ambiente, basato su **Ubuntu 12.04 LTS**, comprendente tutti gli strumenti necessari allo sviluppo del software. Per usufruire di questo ambiente, è però necessario possedere **VMware Player**.

Versione utilizzata: **6.0.1**

5.1.2.1 Installazione: per installare il software, seguire la seguente procedura:

- Collegarsi al sito internet: https://my.vmware.com/web/vmware/free#desktop_end_user_computing/vmware_player/6_0
- Selezionare *Download* in corrispondenza del sistema operativo presente nella propria postazione di lavoro;
- Aprire il terminale;

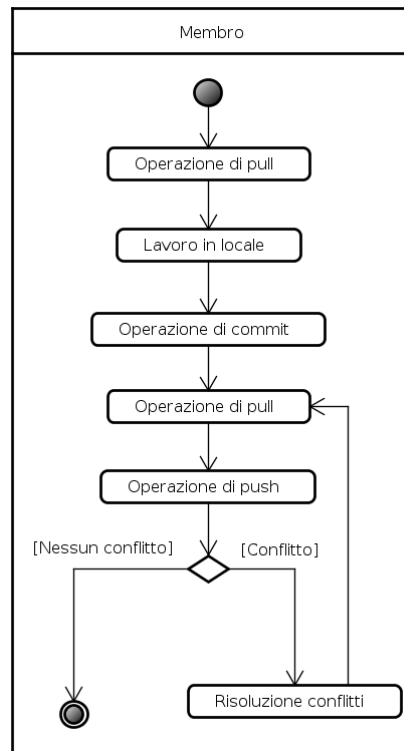


Figura 11: Procedura per l'utilizzo del Repository

- Digitare `sudo apt-get install build-essential linux-headers-$(uname -r)` e procedere con l'installazione;
- Posizionarsi nella directory contenente il file precedentemente scaricato;
- Digitare `gksudo bash nome_del_file.bundle` e seguire la procedura di installazione.

5.2 Strumenti a supporto del processo di documentazione

5.2.1 Astah

Astah è uno strumento multiplatforma che permette di disegnare svariate tipologie di grafici secondo lo standard UML2.0G.

Versione utilizzata: **6.7.0/43495 Professional Edition**

5.2.1.1 Installazione: per l'installazione della versione **Professional** di questo applicativo, è necessario ottenere una licenza per studenti universitari. Seguire la seguente procedura:

- Collegarsi all'indirizzo <http://astah.net/download>
- Selezionare *Download Free Student Edition*. Verrà caricata una pagina contenente una form;
- Compilare la form con i propri dati;
- Selezionare *Send Request*. Verrà inviata una e-mail all'indirizzo indicato e verrà caricata una pagina di ringraziamento;
- Selezionare il proprio sistema operativo e la modalità di installazione che si desidera utilizzare. Verrà scaricato il software;
- Terminato lo scaricamento, procedere con l'installazione;

- Aprire l'e-mail che il sistema ha inviato e selezionare il link presente in essa. Verrà caricata una pagina in cui partirà lo scaricamento della licenza per studenti.
- Aprire **Astah**;
- Selezionare *Tool -> License*;
- Selezionare *Set License Key*. Verrà aperta una finestra di dialog;
- Aprire il file .XML scaricato in precedenza come licenza;
- Selezionare *Select License File*. Il software sarà operativo.

5.2.2 L^AT_EX

Linguaggio di markup_G utilizzato per la stesura della documentazione. Si è deciso di utilizzare la distribuzione T_EXLive 2013, in quanto considerata la più completa e versatile.

5.2.2.1 Installazione: per l'installazione della distribuzione T_EXLive, seguire la seguente procedura:

- Aprire il terminale;
- Digitare: `sudo apt-get install texlive`
- Attendere il completamento dell'installazione.

5.2.3 ReqMonkeys

5.2.3.1 Accesso al sito internet: l'applicazione ReqMonkeys è raggiungibile all'indirizzo:

<http://sevenmonkeys.altervista.org/>

Per utilizzare le funzionalità dell'applicazione, bisogna accedere all'area riservata utilizzando le credenziali di accesso. Le credenziali sono riportate in un documento condiviso in **Google Drive** denominato *Credenziali varie*.

5.2.3.2 Inserimento di un requisito: per inserire un nuovo requisito all'interno del sistema, si segue la seguente procedura:

1. Accedere all'applicazione web (si veda il paragrafo *Accesso al sito internet*);
2. Selezionare *Inserisci requisito*. Verrà caricata una pagina contenente una form per inserire i dati relativi al requisito che si vuole inserire;
3. Compilare la form con tutti i dati necessari;
4. Selezionare il pulsante *Inserisci*.

5.2.3.3 Modificare la fonte di un requisito: per modificare la fonte di un requisito memorizzato nel sistema, si segue la seguente procedura:

1. Accedere all'applicazione web (si veda il paragrafo *Accesso al sito internet*);
2. Selezionare *Modifica fonte requisito*. Verrà caricata una pagina contenente l'elenco di tutti i requisiti memorizzati nel sistema, associati alle relative fonti;
3. Selezionare il requisito di cui si vuole modificare la fonte;
4. Selezionare il pulsante *conferma*. Verrà caricata una pagina contenente le fonti associate al requisito;
5. Selezionare la fonte che si vuole eliminare;
6. Selezionare la nuova fonte da associare al requisito dal menù a tendina;
7. Selezionare il pulsante *Conferma*.

5.2.3.4 Modificare la descrizione e/o l'id di un requisito: per modificare la descrizione e/o l'id di un requisito, si segua la seguente procedura:

1. Accedere all'applicazione web (si veda il paragrafo *Accesso al sito internet*);
2. Selezionare *Modifica descrizione e id requisito*. Verrà caricata una pagina contenente tutti i requisiti memorizzati nel sistema;
3. Selezionare il requisito che si vuole modificare;
4. Selezionare il pulsante *conferma*. Verrà caricata una pagina contenente una form in cui inserire le modifiche da apportare o alla descrizione o all'identificativo del requisito;
5. Compilare la form con le nuove informazioni;
6. Selezionare il pulsante *Conferma*.

5.2.3.5 Aggiungere una fonte ad un requisito: per aggiungere una fonte alla lista di quelle a cui fa riferimento un requisito, seguire la seguente procedura:

1. Accedere all'applicazione web (si veda il paragrafo *Accesso al sito internet*);
2. Selezionare *Aggiungi fonte a requisito*. Verrà caricata una pagina contenente l'elenco dei requisiti memorizzati nel sistema;
3. Selezionare il requisito a cui si vuole aggiungere una fonte;
4. Selezionare il pulsante *Conferma*. Verrà caricata una pagina contenente l'elenco delle fonti memorizzate nel sistema;
5. Selezionare le fonti che si vogliono aggiungere alla lista di quelle a cui fa riferimento il requisito;
6. Selezionare il pulsante *Aggiungi*.

5.2.3.6 Aggiungere un componente ad un requisito: per aggiungere un componente alla lista di quelli a cui fa riferimento un requisito, seguire la seguente procedura:

1. Accedere all'applicazione web (si veda il paragrafo *Accesso al sito internet*);
2. Selezionare *Aggiungi componente a requisito*. Verrà caricata una pagina contenente l'elenco dei requisiti memorizzati nel sistema;
3. Selezionare il requisito a cui si vuole aggiungere un componente;
4. Selezionare il pulsante *Conferma*. Verrà caricata una pagina contenente l'elenco dei componenti memorizzati nel sistema;
5. Selezionare i componenti che si vogliono aggiungere alla lista di quelli a cui fa riferimento il requisito;
6. Selezionare il pulsante *Aggiungi*.

5.2.3.7 Eliminazione di un requisito: per eliminare un requisito dal sistema, si segua la seguente procedura:

1. Accedere all'applicazione web (si veda il paragrafo *Accesso al sito internet*);
2. Selezionare *Elimina requisito*. Verrà caricata una pagina contenente la lista dei requisiti attualmente presenti nel sistema;
3. Selezionare il requisito che si desidera eliminare;
4. Selezionare il pulsante *Elimina*.

5.2.3.8 Rimuovere una fonte da un requisito: per rimuovere una fonte dalla lista di quelle a cui fa riferimento un requisito, seguire la seguente procedura:

- Accedere all'applicazione web (si veda il paragrafo *Accesso al sito internet*);
- Selezionare *Rimuovi fonte da un requisito*. Verrà caricata una pagina contenente l'elenco dei requisiti presenti nel sistema;
- Selezionare il requisito da cui si vuole eliminare una fonte;
- Selezionare il pulsante *Conferma*. Verrà caricata una pagina contenente l'elenco delle fonti associate al requisito;
- Selezionare le fonti che si desiderano eliminare;
- Selezionare il pulsante *Elimina*.

5.2.3.9 Inserimento di una fonte: per inserire una nuova fonte all'interno del sistema, si segua la seguente procedura:

1. Accedere all'applicazione web (si veda il paragrafo *Accesso al sito internet*);
2. Selezionare *Inserisci fonte*. Verrà caricata una pagina contenente la lista delle fonti attualmente presenti nel sistema;
3. Compilare i campi *Codice identificativo* e *Descrizione* assicurandosi che il codice della fonte che si vuole inserire, sia univoco;
4. Selezionare il pulsante *Inserisci*.

5.2.3.10 Eliminazione di una fonte: per eliminare una fonte dal sistema, si segua la seguente procedura:

1. Accedere all'applicazione web (si veda il paragrafo *Accesso al sito internet*);
2. Selezionare *Elimina fonte*. Verrà caricata una pagina contenente la lista delle fonti attualmente presenti nel sistema;
3. Selezionare la fonte che si desidera eliminare;
4. Selezionare il pulsante *Elimina*.

5.2.3.11 Inserimento di un componente: per inserire un nuovo componente all'interno del sistema, si segua la seguente procedura:

1. Accedere all'applicazione web (si veda il paragrafo *Accesso al sito internet*);
2. Selezionare *Inserisci componente*. Verrà caricata una pagina contenente la lista dei componenti presenti nel sistema ed una form che permette di inserire il *Codice identificativo* e la *Descrizione* di un nuovo componente;
3. Compilare la form con le informazioni necessarie, assicurandosi di non inserire un codice identificativo già presente nell'elenco;
4. Selezionare il pulsante *Inserisci*;

5.2.3.12 Rimuovere un componente da un requisito: per rimuovere un requisito dalla lista di quelli a cui fa riferimento un requisito, seguire la seguente procedura:

- Accedere all'applicazione web (si veda il paragrafo *Accesso al sito internet*);
- Selezionare *Rimuovi componente da un requisito*. Verrà caricata una pagina contenente l'elenco dei requisiti presenti nel sistema;
- Selezionare il requisito da cui si vuole eliminare un componente;
- Selezionare il pulsante *Conferma*. Verrà caricata una pagina contenente l'elenco dei componenti associati al requisito;
- Selezionare i componenti che si desiderano eliminare;
- Selezionare il pulsante *Elimina*.

5.2.3.13 Inserire un nuovo componente: per aggiungere un nuovo componente al sistema, seguire la seguente procedura:

- Accedere all'applicazione web (si veda il paragrafo *Accesso al sito internet*);
- Selezionare *Gestione componenti*. Verrà caricata una pagina contenente l'elenco dei componenti presenti nel sistema;
- Selezionare *Inserisci nuovo package*. Verrà caricata una pagina contenente una form che permette di inserire i dati riguardanti un componente;
- Compilare la form con le informazioni necessarie;
- Selezionare il pulsante *Inserisci*;
- Confermare l'inserimento selezionando il pulsante *OK* del pop-up;
- Il sistema non consentirà l'inserimento se la form non è stata interamente completata.

5.2.3.14 Modificare i dettagli di un componente: per modificare i dati relativi ad un componente presente nel sistema, seguire la seguente procedura:

- Accedere all'applicazione web (si veda il paragrafo *Accesso al sito internet*);
- Selezionare *Gestione componenti*. Verrà caricata una pagina contenente l'elenco dei componenti presenti nel sistema;
- Selezionare il componente che si intende modificare;
- Selezionare il pulsante *modifica dettagli*. Verrà caricata una pagina contenente una form compilata con i dati relativi al componente;
- Modificare i campi di interesse;
- Selezionare il pulsante *Modifica*;
- Il sistema non consentirà l'inserimento se la form non è stata interamente completata.

5.2.3.15 Modificare i requisiti associati ad un componente: per modificare le associazioni presenti tra un componente ed i suoi requisiti, seguire la seguente procedura:

- Accedere all'applicazione web (si veda il paragrafo *Accesso al sito internet*);
- Selezionare *Gestione componenti*. Verrà caricata una pagina contenente l'elenco dei componenti presenti nel sistema;
- Selezionare il componente di interesse. Verrà caricata una pagina contenente la lista dei requisiti presenti nel sistema e quelli associati al componente saranno selezionate;
- Selezionare e/o deselectare i requisiti che si vogliono associare e/o dissociare dal componente;
- Selezionare il pulsante *Modifica*.

5.2.3.16 Eliminare un componente dal sistema: per eliminare dal sistema un componente, seguire la seguente procedura:

- Accedere all'applicazione web (si veda il paragrafo *Accesso al sito internet*);
- Selezionare *Gestione componenti*. Verrà caricata una pagina contenente l'elenco dei componenti presenti nel sistema;
- Selezionare il componente che si vuole eliminare;
- Selezionare il pulsante *elimina*;
- Confermare l'eliminazione selezionando il pulsante *OK* del pop-up.

5.2.3.17 Inserire una nuova classe: per aggiungere un nuova classe al sistema, seguire la seguente procedura:

- Accedere all'applicazione web (si veda il paragrafo *Accesso al sito internet*);
- Selezionare *Gestione classi*;
- Selezionare *Inserimento Nuova Classe*. Verrà caricata una pagina contenente una form che permette di inserire i dati riguardanti un componente;
- Compilare la form con le informazioni necessarie;
- Selezionare il pulsante *Inserisci*;
- Confermare l'inserimento selezionando il pulsante *OK* del pop-up;
- Il sistema non consentirà l'inserimento se la form non è stata interamente completata.

5.2.3.18 Modificare i dettagli di una classe: per modificare i dati relativi ad una classe presente nel sistema, seguire la seguente procedura:

- Accedere all'applicazione web (si veda il paragrafo *Accesso al sito internet*);
- Selezionare *Gestione classi*;
- Selezionare *Modifica Classe*. Verrà caricata una pagina contenente una form che permetterà di modificare i dati riguardanti una classe;
- Selezionare il package_G di appartenenza della classe di interesse dal menù a tendina. Verranno caricate le classi contenute nel package_G nel menù a tendina sottostante;
- Selezionare la classe che si vuole modificare dal menù a tendina. Verranno caricate le informazioni riguardanti la stessa, nella form sottostante;
- Modificare i campi di interesse;
- Selezionare il pulsante *Modifica*;
- Il sistema non consentirà l'inserimento se la form non è stata interamente completata.

5.2.3.19 Eliminare una classe dal sistema: per eliminare dal sistema una classe, seguire la seguente procedura:

- Accedere all'applicazione web (si veda il paragrafo *Accesso al sito internet*);
- Selezionare *Gestione classi*;
- Selezionare *Modifica Classe*;
- Selezionare il package_G di appartenenza della classe di interesse dal menù a tendina. Verranno caricate le classi contenute nel package_G nel menù a tendina sottostante;
- Selezionare la classe che si vuole modificare dal menù a tendina. Verranno caricate le informazioni riguardanti la stessa, nella form sottostante;
- Selezionare il pulsante *Elimina*;
- Confermare l'eliminazione selezionando il pulsante *OK* del pop-up.

5.2.3.20 Definire una relazione tra classi: per impostare una relazione tra due classi, seguire la seguente procedura:

- Accedere all'applicazione web (si veda il paragrafo *Accesso al sito internet*);
- Selezionare *Gestione classi*;
- Selezionare *Inserimento Nuova Relazione tra Classi*. Verrà caricata una pagina contenente una form che permette di inserire i dati riguardanti una relazione;
- Compilare la form con le informazioni necessarie;
- Selezionare il pulsante *Inserisci*;
- Confermare l'inserimento selezionando il pulsante *OK* del pop-up;
- Il sistema non consentirà l'inserimento se la form non è stata interamente completata.

5.2.3.21 Modificare una relazione tra classi: per modificare i dati relativi ad una relazione tra classi, seguire la seguente procedura:

- Accedere all'applicazione web (si veda il paragrafo *Accesso al sito internet*);
- Selezionare *Gestione classi*;
- Selezionare *Modifica Relazione tra Classi*;
- Selezionare il package_G di appartenenza della classe di interesse dal menù a tendina. Verranno caricate le classi contenute nel package_G nel menù a tendina sottostante;
- Selezionare la classe di cui si vuole modificare una relazione dal menù a tendina. Verranno caricate le informazioni riguardanti la stessa, nella form sottostante;
- Selezionare la relazione che si vuole modificare;
- Selezionare il pulsante *Modifica*. Verrà caricata una pagina contenente una form popolata con i dati relativi alla relazione;
- Modificare i campi di interesse;
- Selezionare il pulsante *Modifica*;
- Il sistema non consentirà l'inserimento se la form non è stata interamente completata.

5.2.3.22 Eliminare una relazione tra classi: per eliminare i dati relativi ad una relazione tra classi, seguire la seguente procedura:

- Accedere all'applicazione web (si veda il paragrafo *Accesso al sito internet*);
- Selezionare *Gestione classi*;
- Selezionare *Modifica Relazione tra Classi*;
- Selezionare il package_G di appartenenza della classe di interesse dal menù a tendina. Verranno caricate le classi contenute nel package_G nel menù a tendina sottostante;
- Selezionare la classe di cui si vuole eliminare una relazione dal menù a tendina. Verranno caricate le informazioni riguardanti la stessa, nella form sottostante;
- Selezionare la relazione che si vuole eliminare;
- Selezionare il pulsante *Elimina*;
- Confermare l'eliminazione selezionando il pulsante *OK* del pop-up.

5.2.3.23 Gestire i metodi, gli attributi e le note delle classi: l'applicazione **reqMonkeys** permette di poter gestire tutte le entità associate alle classi del progetto. Per poter inserire o eliminare tali entità (metodi, attributi e note), seguire la seguente procedura:

- Accedere all'applicazione web (si veda il paragrafo *Accesso al sito internet*);
- Selezionare *Gestione classi*;
- Selezionare *Gestione proprietà*. Verrà caricato un menù a tendina in cui è possibile selezionare il package_G in cui si vuole operare;
- Selezionare il package_G contenente la classe di cui si vogliono modificare le proprietà. Verrà caricata una tabella contenente tutte le classi appartenenti al package_G memorizzate nel sistema;
- Selezionando i pulsanti *show* relativi alla classe desiderata, è possibile visualizzare gli attributi ed i metodi associati ad essa. Inoltre, selezionando i pulsanti *Aggiungi*, è possibile inserire i dati relativi ad un nuovo attributo o ad un nuovo metodo della classe. Completare i campi della form presenti nel pop-up e confermare selezionando *OK*;
- Dalle tabelle contenenti gli attributi ed i metodi di una classe, è possibile eliminare gli stessi, utilizzando i pulsanti *Elimina*;
- Infine, è possibile gestire le note associate a metodi e attributi, utilizzando i pulsanti *Aggiungi* ed *Elimina* nelle rispettive tabelle.

5.2.3.24 Gestire i tipi di Romeo: per inserire e/o eliminare i tipi di dati presenti nel software, seguire la seguente procedura:

- Accedere all'applicazione web (si veda il paragrafo *Accesso al sito internet*);
- Selezionare *Gestione tipi*. Verrà caricata una pagina contenente la lista dei tipi contenuti nel sistema;
- Per inserirne uno nuovo, selezionare il pulsante *Inserisci tipo*, completare la form e confermare con il pulsante *OK*;
- Per eliminare un tipo esistente, selezionare il relativo pulsante *Elimina* e confermare selezionando *OK*.

5.2.4 T_EXMaker

Editor di testo L^AT_EXmulti piattaforma, che consente di compilare i file .TEX ed eventualmente di generare i file .PDF.

Versione utilizzata: **4.0.3**

5.2.4.1 Installazione: per l'installazione dell'applicativo, seguire la seguente procedura:

- Aprire il terminale;
- Digitare `sudo apt-get install texmaker`;
- Aspettare il completamento dell'installazione.

5.2.5 Visual Paradigm

Software che permette la creazione di svariati diagrammi nel linguaggio UML, conforme allo standard 2.0. Tale software offre i seguenti vantaggi:

- Distinzione tra i diagrammi delle classi e package_G;
- Il software è multipiattaforma;

- Generazione automatica del codice dello “scheletro” delle classi;
- Facilità di utilizzo del software.

Versione utilizzata: **11.0 Community Edition**

5.2.5.1 Installazione: per l'installazione dell'applicativo, seguire la seguente procedura:

- Collegarsi al sito internet: <http://www.visual-paradigm.com/download/vpuml.jsp?edition=ce>
- Selezionare il proprio sistema operativo;
- Selezionare *Get Community Edition*. Verrà scaricato un file `.SH`
- Aprire il terminale e posizionarsi nella directory in cui è stato scaricato il file precedente;
- Digitare: `bash nome_del_file.sh` ed attendere il completamento dell'installazione.

5.3 Strumenti a supporto del processo di verifica

5.3.1 Aspell

GNU Aspell è un correttore ortografico Open Source. Può essere utilizzato sia come singola libreria, sia come correttore indipendente. A differenza del correttore integrato in `TeXmaker` (vedi 5.2.4), **Aspell** suggerisce un termine da inserire al posto di una parola errata ed inoltre consente di memorizzare dei termini non presenti nel dizionario.

Versione utilizzata: **3.1.20**

5.3.1.1 Installazione: per l'installazione dell'applicativo, seguire la seguente procedura:

- Aprire il terminale;
- Digitare il comando `sudo apt-get install aspell-it`;
- Seguire la procedura di installazione fino al termine.

5.3.1.2 Eseguire un controllo ortografico: per eseguire un controllo su un file di testo (in questo caso come esempio prendiamo un file `.TEX`), seguire la seguente procedura:

- Aprire il terminale;
- Posizionarsi nella directory in cui è presente il file da processare;
- Digitare il comando `aspell --mode=tex --lang=it check nomedocumento.tex`;
- Procedere con il controllo ortografico seguendo le indicazioni dell'applicativo.

5.3.2 Mantis

Applicazione web che permette di tenere traccia dei bug individuati nel software. Tale servizio è stato integrato nel sito internet del team ed è accessibile a tutti i membri del gruppo tramite username e password personali.

5.3.2.1 Accesso: per accedere al servizio, seguire la seguente procedura:

- Collegarsi al sito internet del team: <http://sevenmonkeys.altervista.org/>
- Selezionare *Area Riservata* ed inserire le credenziali;
- Selezionare *Bug Trucker*;
- Se necessario, inserire le proprie credenziali per accedere a **Mantis** (in fig.12 viene mostrata la schermata di login di Mantis).



Login	
Username	<input type="text"/>
Password	<input type="password"/>
Remember my login in this browser	<input type="checkbox"/>
Secure Session	<input checked="" type="checkbox"/> Only allow your session to be used from this IP address.
<input type="button" value="Login"/>	

[[Signup for a new account](#)] [[Lost your password?](#)]

Figura 12: Schermata di login di Mantis

5.4 Strumenti a supporto del processo di gestione del progetto

5.4.1 Dropbox

Dropbox è un servizio di cloud che consente di condividere directory e file tra più utenti. Verrà creata una cartella *Ingegneria del Software* condivisa da tutti i membri del gruppo. La presenza del client ufficiale disponibile per la maggior parte dei sistemi operativi ha fatto cadere la scelta su questo servizio cloud.

Versione utilizzata: **2.4.11**

5.4.1.1 Installazione: per l'installazione dei client **Dropbox** seguire la seguente procedura:

- Collegarsi al sito internet <https://www.dropbox.com/install?os=linux>
- Scegliere la propria distribuzione e la tipologia di architettura del proprio pc. Verrà scaricato un file con cui si potrà procedere all'installazione;
- Installare il software seguendo la procedura;
- Aprire il software e seguire la procedura per creare un nuovo account o per effettuare il login con le proprie credenziali.

5.4.2 GanttProject

GanttProject è un software Open Source per la gestione di attività di progetto. La scelta è ricaduta su tale applicativo per i seguenti motivi:

- Portabilità, essendo il software scritto in Java;
- Open source;
- Permette la generazione di diagrammi di Gantt_G e delle risorse;
- Consente di esportare i diagrammi in formato PNG o HTML.

Versione utilizzata: **2.6.5**

5.4.2.1 Installazione: per l'installazione dell'applicativo, seguire la seguente procedura:

- Collegarsi al sito internet <http://www.ganttproject.biz/>
- Selezionare *Download GanttProject 2.6.5*. Verrà caricata una pagina in cui sarà presente il collegamento adatto al proprio sistema operativo;
- Selezionare *GanttProject 2.6.5*. Verrà caricata una pagina contenente le ultime versioni disponibili del software;

- Selezionare l'ultima versione. Verrà caricata un'ulteriore pagina di download;
- Selezionare il file da scaricare. Partirà automaticamente il download del file;
- Una volta scaricato, procedere con l'installazione.

5.4.3 GitHub

GitHub è un servizio web di hosting per lo sviluppo di progetti, che utilizza il controllo di versione **Git**. Sono stati creati due repository privati, sfruttando il servizio di **GitHub Education**³⁴:

- Documentazione: <https://github.com/nicolobissacco/7MonkeysDoc>
- Codice: <https://github.com/nicolobissacco/7MonkeysCode>

Rendere operativi gli account dei membri del team, collegandoli ai repository, sarà compito degli *Amministratori*.

La scelta è ricaduta su **Git** per i seguenti motivi:

- La possibilità di poter lavorare localmente, senza bisogno di essere connessi a internet;
- **Git** era già stato usato da vari membri del team;
- La possibilità di ignorare certe estensioni dei file (gitignore);
- La presenza di innumerevoli client grafici, per chi non volesse utilizzarlo da shell.

Per i membri del gruppo, che non vogliono utilizzare la shell, sono consigliati i seguenti client:

- **Sourcetree**: disponibile per i sistemi Windows e Mac OS;
- **Giteye**: disponibile per i sistemi Linux.

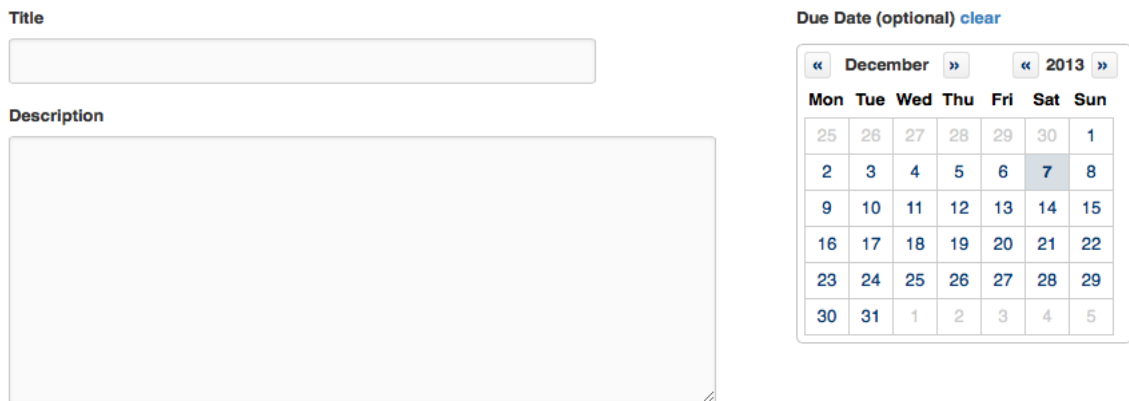
5.4.3.1 Registrazione: per utilizzare il servizio, è necessario registrarsi presso il portale di **GitHub**. Seguire la seguente procedura:

- Collegarsi al sito internet <https://github.com/>
- Compilare la form e selezionare *Sign up for GitHub*;
- Seguire la procedura indicata nella e-mail che viene inviata all'indirizzo indicato nella form.

5.4.3.2 Creazione milestone: per creare una nuova milestone, si segua la seguente procedura:

- Accedere al repository, collegandosi all'indirizzo web opportuno (vedi sez.5.4.3);
- Selezionare *Issues*. Verrà caricata una pagina contenente la lista dei ticket attualmente aperti;
- Selezionare *Milestones*. Verrà caricata una pagina contenente le milestone create;
- Selezionare *Create a new milestone*. Verrà caricata una pagina contenente una form (vedi 13);
- Compilare i campi della form;
- Selezionare *Create milestone*.

³⁴<https://education.github.com/>



Title

Description

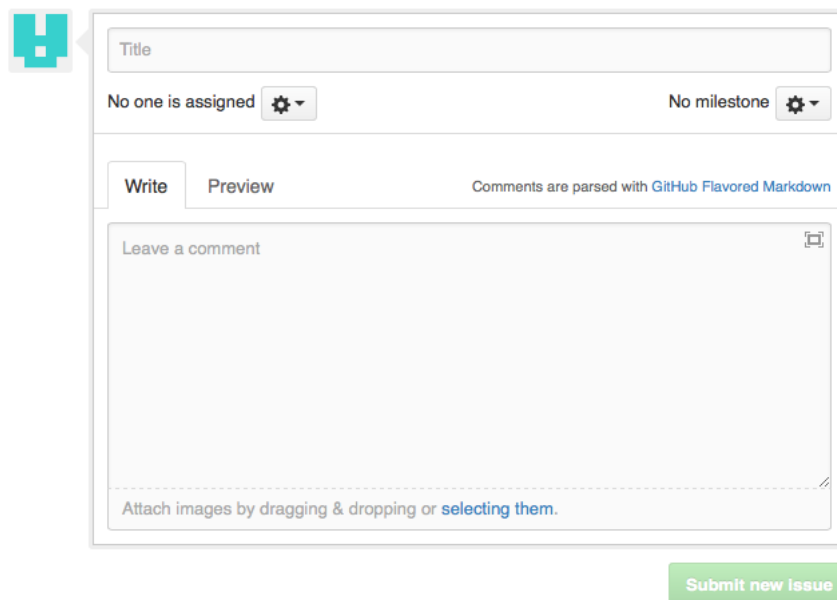
Due Date (optional) clear

« December »		« 2013 »				
Mon	Tue	Wed	Thu	Fri	Sat	Sun
25	26	27	28	29	30	1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30	31	1	2	3	4	5



Figura 13: Interfaccia di GitHub per la creazione di una milestone

5.4.3.3 Creazione ticket: per creare un nuovo ticket, si segua la seguente procedura:


- Accedere al repository_G, collegandosi all'indirizzo web opportuno (vedi sez.5.4.3);
- Selezionare *Issues*. Verrà caricata una pagina contenente la lista dei ticket attualmente aperti;
- Selezionare *New Issue*. Verrà caricata una pagina contenente una form (vedi figura 14);
- Compilare la form assegnando: titolo, descrizione, milestone, membro destinatario del ticket ed eventuali *labels*;
- Selezionare *Submit new Issue*.



Title

No one is assigned  No milestone 

Write **Preview** Comments are parsed with [GitHub Flavored Markdown](#)

Leave a comment 

Attach images by dragging & dropping or [selecting them](#).

Submit new Issue

Figura 14: Interfaccia di GitHub per la creazione di un ticket.

5.4.3.4 Chiusura milestone: per chiudere la milestone_G, si segua la seguente procedura:

- Accedere al repository_G, collegandosi all'indirizzo web opportuno (vedi sez.5.4.3);
- Selezionare *Issues*. Verrà caricata una pagina contenente la lista dei ticket attualmente aperti;
- Selezionare *Milestones*. Verrà caricata una pagina contenente le milestone_G create;
- Selezionare *Close* in corrispondenza della milestone_G che si vuole chiudere.

5.4.4 Google Calendar

Per sfruttare le potenzialità di questo strumento collaborativo, è stato creato un calendario condiviso tra i vari membri del gruppo denominato **GruppoSWE**. Tale servizio permette di notificare:

- In quali date un certo membro non è disponibile;
- In quali date c'è una riunione o un evento rilevante ai fini del gruppo.

Grazie alla gestione delle notifiche di **Google Calendar_G**, è possibile far in modo che 24 ore prima di un evento rilevante venga inviata una email a tutti i membri del gruppo come promemoria.

5.4.4.1 Registrazione: per usufruire di questo servizio, è necessario possedere un account **Google**. Sarà cura degli *Amministratori*, di condividere il calendario con tutti i membri del gruppo. Per ottenere un account **Google**, seguire la seguente procedura:

- Collegarsi al sito internet: <https://accounts.google.com/SignUp?hl=it>
- Compilare la form con i propri dati;
- Selezionare *Passaggio successivo*;
- Seguire la procedura indicata nella e-mail di conferma che verrà inviata all'indirizzo precedentemente inserito.

5.4.4.2 Creare un nuovo evento: per creare un nuovo evento, seguire la seguente procedura:

- Collegarsi al sito internet: <https://www.google.com/calendar/>
- Se necessario, effettuare l'accesso al sistema di **Google** inserendo le proprie credenziali;
- Selezionare la freccia a fianco al nome del calendario;
- Selezionare *Crea evento in questo calendario*;
- Compilare la form con tutti i dati necessari;
- Selezionare *Salva*.

5.4.5 Google Drive

Servizio di cloud storage, che permette di immagazzinare e condividere file con più persone. Viene utilizzato dal team per organizzare i file sviluppati tramite il servizio **Google Docs_G**. Per questo scopo è stata creata e condivisa una cartella denominata **swe**.

5.4.5.1 Registrazione: per usufruire del servizio è necessario possedere un account **Google**. Per la procedura, si veda il paragrafo *Registrazione* nella sezione ??.

5.4.5.2 Creazione di un nuovo documento condiviso: per creare un nuovo documento condiviso e salvarlo nella cartella condivisa, seguire la seguente procedura:

- Collegarsi al sito internet: <https://drive.google.com/>
- Se necessario, effettuare l'accesso al sistema di **Google** inserendo le proprie credenziali;
- Selezionare *Shared with Me*;
- Selezionare la cartella *swe*;
- Selezionare *Create* e seguire la procedura di creazione di un nuovo documento condiviso.

5.4.6 Scripts

Per facilitare la compilazione e il controllo dei documenti, è stato creato un *Makefile* disponibile nel repository **G** della documentazione. (vedi fig. 9). Il *Makefile* consente di:

- Generare tutti i file della documentazione in formato **.PDF**
- Eseguire il controllo ortografico in tutti i file **.TEX** che compongono la struttura della documentazione;
- Eliminare i file generati dalla compilazione dei file **.TEX**

5.4.6.1 Generare la documentazione: per compilare la documentazione, in occasione di una consegna di revisione, seguire la seguente procedura:

- Aprire il terminale;
- Posizionarsi nella cartella *Scripts* (vedi fig. 9);
- Digitare **make RX** con **X** corrispondente alla revisione che si sta per consegnare (**R** per Revisione, **P** per Progettazione, **Q** per Qualifica, **A** per Accettazione). Verranno generati tutti i file e verranno salvati nell'apposita directory *Consegne*.

5.4.6.2 Eseguire controllo ortografico: per eseguire il controllo ortografico, seguire la seguente procedura:

- Aprire il terminale;
- Posizionarsi nella cartella *Scripts* (vedi fig. 9);
- Digitare **make aspell**. Verrà avviata la procedura di controllo ortografico, utilizzando **Aspell** (vedi sez. 5.3.1). Il controllo verrà effettuato su tutti i file **.TEX** che compongono la struttura della documentazione.

5.4.6.3 Pulire il filesystem: per eseguire la rimozione dei file di compilazione della documentazione, seguire la seguente procedura:

- Aprire il terminale;
- Posizionarsi nella cartella *Scripts* (vedi fig. 9);
- Digitare **make clean**. Verranno rimossi tutti i file di compilazione generati da **L^AT_EX**.

A Lista di controllo

Nella verifica dei documenti, si sono rilevati, più frequentemente, errori dovuti a:

- **Non aderenza delle Norme di Progetto:** alcuni errori ricorrenti sono dovuti alla non piena conoscenza, da parte dei redattori, delle norme riguardanti la stesura dei documenti (vedi sezione 3.1.2.5) in particolare:
 - Inserito il “.” oppure nessun simbolo di punteggiatura al posto del “;” negli elenchi puntati al termine di ogni voce interna;
 - Errato utilizzo del markup nei riferimenti delle estensioni ai file;
 - Inserimento lettera maiuscola su titoli paragrafi ove non necessario;
 - Mancanza, nei termini di glossario, della marcatura “G” in pedice;
 - Non utilizzato il corsivo per il riferimento ad altri documenti;
 - Mancato utilizzo dei comandi creati per alcuni termini di uso ricorrente.
- **Errori ortografici:**
 - Eccessivo utilizzo del punto e virgola nei periodi. È preferibile inserire il punto;
 - Dovuti a distrazioni e/o errori di battitura;
 - Punteggiatura sbagliata o mancante.
- **Errori riguardanti la lingua italiana:**
 - Passaggio da un tempo verbale ad un altro nello stesso periodo;
 - Utilizzo di parole il cui significato attribuito dal redattore, non è conforme a quello dato dalla lingua italiana;
 - Inserimento della virgola tra soggetto e verbo. Rende poco comprensibile il periodo.
- **Errori riguardanti la stesura dei diagrammi UML:**
 - Nei diagrammi delle classi, le classi e i metodi astratti non vengono riportati in corsivo;
 - Nei diagrammi delle classi non viene utilizzata l’esatta relazione di implementazione di un’interfaccia;
 - Nei diagrammi delle classi è stata utilizzata la relazione d’uso invece della relazione di creazione;
 - Mancata coerenza tra nomi presenti nei documenti e nomi nei diagrammi;
 - Nei digrammi delle classi è stata utilizzata erroneamente la relazione di aggregazione invece che quella di composizione;
 - Nei digrammi di sequenza mancato inserimento del messaggio di ritorno.