



CLIPS

Norme di Progetto v0.1

Sommario

Questo documento definisce le norme stabilite dal gruppo Beacon Strips per la realizzazione del progetto CLIPS

Nome del documento	Norme di Progetto
Versione	0.1
Data di redazione	2016/04/04
Redazione	Matteo Franco Luca Soldera
Verifica	Enrico Bellio
Approvazione	Viviana Alessio
Uso	Interno
Lista di distribuzione	prof. Tullio Vardanega prof. Riccardo Cardin Miriade SpA

Diario delle modifiche

Versione	Riepilogo	Autore	Ruolo	Data
0.5.2	Correzione sezione Repository relativa al Processo di gestione)	Luca Soldera	Amministratore	2016-03-09
0.5.1	Correzione sezione Norme relativa al Processo di documentazione	Matteo Franco	Amministratore	2016-03-09
0.6.0	Stesura sezioni Procedure e Norme relative al Processo di gestione	Luca Soldera	Amministratore	2016-03-08
0.5.0	Stesura sezioni Attività e Strumenti relative al Processo di gestione	Matteo Franco	Amministratore	2016-03-07
0.4.0	Stesura sezione Processi primari	Luca Soldera	Amministratore	2016-03-06
0.3.0	Completamento stesura sezione Processo di documentazione	Matteo Franco	Amministratore	2016-03-06
0.2.1	Stesura sezione Processo di verifica	Luca Soldera	Amministratore	2016-03-04
0.2.0	Inizio stesura sezione Processo di documentazione	Matteo Franco	Amministratore	2016-03-04
0.1.0	Creazione scheletro documento e stesura sezione Introduzione	Matteo Franco	Amministratore	2016-03-03

Indice

1	Introduzione	4
1.1	Scopo del documento	4
1.2	Scopo del prodotto	4
1.3	Glossario	4
1.4	Riferimenti	4
1.4.1	Riferimenti informativi	4
2	Processi primari	5
2.1	Processo di sviluppo	5
2.1.1	Attività	5
2.1.1.1	Analisi dei requisiti	5
2.1.1.1.1	Studio di Fattibilità e Analisi dei Rischi	5
2.1.1.1.2	Analisi dei requisiti	5
2.1.2	Norme	5
2.1.2.1	Classificazione dei requisiti	5
2.1.2.2	Classificazione dei casi d'uso	6
2.1.3	Strumenti	6
2.1.3.1	Trender	6
3	Processi di supporto	6
3.1	Processo di documentazione	6
3.1.1	Procedure	6
3.1.1.1	Ciclo di vita	6
3.1.2	Norme	7
3.1.2.1	Template	7
3.1.2.2	Norme tipografiche	7
3.1.2.2.1	Stile del testo	7
3.1.2.2.2	Punteggiatura	8
3.1.2.2.3	Composizione del testo	8
3.1.2.2.4	Formati	8
3.1.2.3	Componenti grafiche	9
3.1.2.3.1	Immagini	9
3.1.2.3.2	Tabelle	9
3.1.2.4	Struttura del documento	9
3.1.2.4.1	Frontespizio	9
3.1.2.4.2	Diario delle modifiche	9
3.1.2.4.3	Indici	10
3.1.2.4.4	Intestazione e piè di pagina	10
3.1.2.5	Versionamento	10
3.1.2.6	Strumenti	11
3.1.2.6.1	L ^A T _E X	11
3.1.2.6.2	Correttore ortografico	11
3.1.2.6.3	Script	11
3.2	Processo di verifica	11
3.2.1	Attività	11
3.2.1.1	Verifica di un ticket	11
3.2.2	Procedure	11
3.2.2.1	Gestione delle anomalie	11
4	Processi organizzativi	13
4.1	Processo di gestione	13
4.1.1	Attività	13
4.1.1.1	Comunicazioni	13

4.1.1.1.1	Comunicazioni interne	13
4.1.1.1.2	Comunicazioni esterne	13
4.1.1.2	Riunioni	13
4.1.1.2.1	Riunioni interne	13
4.1.1.2.2	Riunioni esterne	14
4.1.1.3	Ticket	14
4.1.1.3.1	Lista dei tag	14
4.1.2	Procedure	14
4.1.2.1	Creazione di un ticket	14
4.1.2.2	Aggiornamento di un ticket	16
4.1.3	Norme	16
4.1.3.1	Repository	16
4.1.3.1.1	Struttura del repository	16
4.1.3.1.2	Tipi di file e .gitignore	16
4.1.3.1.3	Norme sulla commit	16
4.1.3.1.4	Messaggi delle commit	18
4.1.4	Strumenti	18
4.1.4.1	Sistema operativo	18
4.1.4.2	Condivisione	18
4.1.4.3	Gestione	18

1 Introduzione

1.1 Scopo del documento

Questo documento intende definire le norme da rispettare all'interno del gruppo Beacon Strips durante lo sviluppo del progetto CLIPS.

Tutti i membri del team sono tenuti a visionare il documento e a rispettare le norme. Tali norme permettono di ottenere uniformità nei documenti sviluppati, migliorare l'efficienza del lavoro svolto e ridurre il numero di errori.

In particolare si tratteranno:

- le interazioni tra i membri del team;
- le interazioni del team con componenti esterne;
- le modalità di stesura dei documenti;
- la gestione del *repository*_g;
- le modalità di lavoro durante le varie fasi del progetto;
- l'ambiente di lavoro utilizzato.

1.2 Scopo del prodotto

Lo scopo del prodotto è la ricerca di nuovi scenari riguardanti la navigazione tramite l'utilizzo dei *beacon*_g. Inoltre verrà realizzata un'applicazione per *smartphones*_g che permette di interfacciarsi con i beacon.

1.3 Glossario

Al fine di evitare ogni ambiguità nel linguaggio e massimizzare la comprensione dei documenti, i termini tecnici, gli acronimi e le abbreviazioni che necessitano di definizione sono riportati nel documento "*Glossario v1.0.0*".

Inoltre ogni occorrenza di un vocabolo presente nel Glossario sarà posta in corsivo e seguita da una 'g' minuscola a pedice (p.es. *Glossario*_g).

1.4 Riferimenti

1.4.1 Riferimenti informativi

- ISO 8601: <http://www.iso.org/iso/home/standards/iso8601.htm>;
- GitHub: <http://github.com/>.

2 Processi primari

2.1 Processo di sviluppo

2.1.1 Attività

2.1.1.1 Analisi dei requisiti

Sarà compito degli analisti redigere l'analisi dei requisiti in seguito a riunioni interne col team e allo studio dei capitolati d'appalto.

2.1.1.1.1 Studio di Fattibilità e Analisi dei Rischi

Lo studio di fattibilità sarà il primo documento dell'analisi dei requisiti che analizzerà i seguenti punti per ogni capitolato, con maggior interesse per quello scelto dal team:

- **Scopo del progetto:** analisi delle richieste del capitolato;
- **Studio del dominio:** valutazione delle tecnologie e conoscenze richieste rispetto l'attuale livello del team;
- **Analisi dei rischi:** ricerca dei rischi e delle criticità per ciascun capitolato.

2.1.1.1.2 Analisi dei requisiti

Il secondo documento che gli analisti andranno a redigere sarà l'analisi dei requisiti che produrrà dei requisiti semplici a partire dalle informazioni raccolte tramite lo studio del capitolato e riunioni esterne con il proponente. Per rendere più precisa e veloce la stesura dei requisiti è stato utilizzato il software *Trender_g*.

2.1.2 Norme

2.1.2.1 Classificazione dei requisiti

I requisiti saranno rappresentati secondo la seguente codifica:

R[importanza][tipo][identificativo]

- **Importanza:** indica se il requisito è:
 1. **0:** obbligatorio;
 2. **1:** desiderabile;
 3. **2:** opzionale.
- **Tipo:** indica se è di tipo:
 1. **F:** funzionale;
 2. **Q:** di qualità;
 3. **P:** prestazionale;
 4. **V:** di vincolo.
- **Identificativo:** è il codice univoco e gerarchico che automaticamente il software assegna al requisito (esempio: 4.2.1);
- **Descrizione:** una breve descrizione del requisito;

- **Fonte:** la fonte da cui deriva il requisito.

2.1.2.2 Classificazione dei casi d'uso

I casi d'uso saranno rappresentati secondo la seguente codifica:

UC[identificativo]

- **Identificativo:** codice univoco e gerarchico che automaticamente il software assegna al caso d'uso (esempio: 3.4.1);

inoltre i casi d'uso saranno caratterizzati da:

- **Tipo:** se non specificato è di tipo standard altrimenti viene scelto fra:
 1. **estensione;**
 2. **inclusione;**
 3. **generalizzazione.**
- **titolo** breve del caso d'uso;
- **descrizione** del caso d'uso;
- **precondizione** del caso d'uso;
- **postcondizione** del caso d'uso
- **scenario principale** degli eventi;
- **scenario secondario** eventuale;
- **attori:** lista degli attori coinvolti;

2.1.3 Strumenti

2.1.3.1 Trender

Per velocizzare e automatizzare la stesura dei requisiti e dei casi d'uso è stato utilizzato il software **Trender** sviluppato da Simone Campagna del gruppo InfiniTech dell'anno 2014/2015. Il software permette di tracciare i requisiti e i casi d'uso assegnando automaticamente un codice univoco che rispetti la sintassi desiderata.

3 Processi di supporto

3.1 Processo di documentazione

3.1.1 Procedure

3.1.1.1 Ciclo di vita

Ogni documento può trovarsi in tre fasi differenti:

- **In lavorazione:** un documento è in lavorazione quando vengono aggiunti, modificati o rimossi dei contenuti;
- **Da verificare:** quando un documento è completo dovrà essere preso in consegna dai verificatori, che si occuperanno di rilevare e/o correggere errori sintattici e semantici;

- **Approvato:** un documento, ultimata la verifica, deve essere approvato dal Responsabile. L'approvazione determina lo stato finale della versione del documento.

Le varie fasi vengono scandite dal sistema di ticketing (vedi sezione)

3.1.2 Norme

3.1.2.1 Template

È presente una cartella nel *repository*_g in `/documenti/template`.

La cartella contiene i seguenti files:

- *Comandi.sty*: contiene i comandi personalizzati (ad es. `\g1`);
- *Riferimenti.sty*: all'interno del file sono presenti vari comandi da utilizzare come scorciatoie (ad es. `\RES`);
- *Stile.sty*: contiene gli stili grafici da applicare al template;
- *TemplateDoc.tex*: è il *template*_g da utilizzare per realizzare i documenti. Ogni documento dovrà essere realizzato a partire da questo file;
- *Glossario.sty*: questo file serve per impostare il *Glossario*, che avrà una configurazione diversa al template standard.

È presente inoltre una cartella *img* dove si trova il logo del team e dove potranno essere caricate tutte le altre immagini necessarie al template.

3.1.2.2 Norme tipografiche

In questa sezione vengono definite le norme riguardanti l'ortografia e la tipografia, al fine di avere uno stile uniforme per tutti i documenti prodotti.

3.1.2.2.1 Stile del testo

- **Grassetto:** il grassetto va utilizzato nei seguenti casi:
 - titoli;
 - elenchi puntati: può essere utilizzato il grassetto nel caso sia necessario evidenziare il concetto da sviluppare;
 - altri casi: per evidenziare parole chiave o contenuti importanti.
- **Corsivo:** lo stile corsivo si applica a:
 - documenti;
 - abbreviazioni.
- **L^AT_EX:** ogni occorrenza di L^AT_EX va scritta con il comando `\LaTeX`
- **Maiuscolo:** è possibile utilizzare lo stile maiuscolo **solo** per gli acronimi;
- **Monospace_g:** le porzioni di testo scritte in *monospace*_g definiscono:
 - frammenti di codice;
 - comandi;
 - URL.
- **Glossario:** le parole che hanno un riferimento nel glossario sono in corsivo e hanno una 'g' a pedice.

3.1.2.2.2 Punteggiatura

- **Spaziature:** ogni simbolo di punteggiatura deve essere seguito da uno spazio, e mai preceduto;
- **Parentesi:** le parentesi costituiscono un'eccezione riguardo le spaziature, in quanto devono essere precedute da uno spazio, ma non seguite;
- **Virgolette singole:** le virgolette singole indicano un singolo carattere;
- **Virgolette doppie:** le virgolette doppie vanno utilizzate per indicare citazioni e documenti.

3.1.2.2.3 Composizione del testo

- **Elenchi puntati:** ogni punto dell'elenco termina con un punto e virgola, ad eccezione dell'ultimo che deve terminare con un punto fermo.
Ogni punto inizia con la minuscola, tranne nel caso in cui necessiti di una spiegazione: allora si utilizzerà la maiuscola;
- **Nota a piè di pagina:** ogni nota inizia con l'iniziale della prima parola maiuscola e termina con il punto. Non ci sono spaziature tra il numero della nota e il testo.

3.1.2.2.4 Formati

- **Date:** per le date va utilizzata la notazione definita dallo standard *ISO_g 8601:2004*:

$$AAAA - MM - GG$$

dove:

- AAAA: rappresenta l'anno utilizzando quattro cifre;
- MM: rappresenta il mese utilizzando due cifre;
- GG: rappresenta il giorno utilizzando quattro cifre;
- **Sigle:** le sigle dei documenti vanno utilizzate solo nei diagrammi o nelle tabelle, con lo scopo di risparmiare spazio. Tali sigle sono:
 - **AdR** per “*Analisi dei Requisiti v1.0.0*”;
 - **G1** per “*Glossario v1.0.0*”;
 - **NdP** per “*Norme di Progetto v1.0.0*”;
 - **PdP** per “*Piano di Progetto v1.0.0*”;
 - **PdQ** per “*Piano di Qualifica v1.0.0*”;
 - **SdF** per “*Studio di Fattibilità v1.0.0*”;
 - **ST** per “*Specifica Tecnica v1.0.0*”.
- **Ruoli di progetto:** per indicare i vari ruoli vanno utilizzati i seguenti comandi:
 - **\AM** per Amministratore;
 - **\AN** per Analista;
 - **\PR** per Programmatore;
 - **\PRJ** per Progettista;
 - **\RES** per Responsabile;
 - **\VER** per Verificatore.

- **Nomi propri:** per indicare nomi propri va utilizzata la forma *Nome Cognome*;
- **Nome del proponente:** tramite il comando \PROPONENTE ci si riferisce al proponente Miriade SpA;
- **Nome del progetto:** con \PROGETTO verrà citato il nome del progetto, ovvero CLIPS;
- **Nome del committente:** \COMMITTENTE viene impiegato per riferirsi al Prof. Tullio Vardanega.

3.1.2.3 Componenti grafiche

3.1.2.3.1 Immagini

Il formato preferito per le immagini è SVG (*Scalable Vector Graphics_g*), in quanto è possibile garantire una qualità maggiore e sono ridimensionabili senza perdere qualità. Nel caso non sia possibile integrare questo formato, si raccomanda l'utilizzo di immagini in formato PNG (*Portable Network Graphics_g*).

3.1.2.3.2 Tabelle

Le tabelle devono essere accompagnate da una didascalia e da un numero incrementale per garantirne la tracciabilità.

3.1.2.4 Struttura del documento

3.1.2.4.1 Frontespizio

La prima pagina di ogni documento dovrà contenere le seguenti informazioni:

- nome del gruppo;
- nome del progetto;
- nome del documento e la sua versione;
- sommario;
- data di redazione;
- nome e cognome dei redattori del documento;
- nome e cognome dei verificatori del documento;
- nome e cognome del responsabile per l'approvazione del documento;
- destinazione d'uso del documento;
- lista di distribuzione del documento.

3.1.2.4.2 Diario delle modifiche

La seconda pagina di ogni documento dovrà contenere il diario delle modifiche.

Il diario consiste in una tabella ordinata in modo decrescente secondo la data di modifica e, conseguentemente, al numero di versione. È particolarmente utile per tenere traccia delle varie modifiche effettuate nel documento.

Ogni riga del diario conterrà:

- numero di versione;
- breve riepilogo delle modifiche effettuate;

- autore delle modifiche;
- ruolo ricoperto dall'autore;
- data di modifica.

3.1.2.4.3 Indici

In ogni documento si trova un indice delle sezioni, il quale fornisce una visione macroscopica della struttura del documento. Nel caso comparissero tabelle e figure nel documento, potranno essere presenti i rispettivi indici.

3.1.2.4.4 Intestazione e piè di pagina

L'intestazione di ogni pagina contiene i seguenti elementi:

- numero della sezione;
- titolo della sezione.

A piè di pagina invece si trovano:

- nome del documento e numero di versione, allineato a sinistra;
- nome del team, allineato al centro;
- pagina X di Y, dove X è la pagina corrente e Y è il numero di pagine totali, allineato a destra.

3.1.2.5 Versionamento

I documenti prodotti sono soggetti al seguente versionamento attraverso la seguente codifica:

vX.Y.Z

dove:

- **X**: indica il numero crescente di uscite formali del documento;
- **Y**: indica il numero crescente di modifiche sostanziali al documento quali stesura, verifica, approvazione;
- **Z**: indica il numero crescente di modifiche minori effettuate sul documento.

I documenti vanno citati sempre con una versione specifica del documento, utilizzando i comandi:

- \ARdoc per il documento “*Analisi dei Requisiti v1.0.0*”;
- \Gldoc per il documento “*Glossario v1.0.0*”;
- \NPdoc per il documento “*Norme di Progetto v1.0.0*”;
- \PPdoc per il documento “*Piano di Progetto v1.0.0*”;
- \PQdoc per il documento “*Piano di Qualifica v1.0.0*”;
- \SFdoc per il documento “*Studio di Fattibilità v1.0.0*”;
- \STdoc per il documento “*Specifiche Tecnica v1.0.0*”.

Il formato da utilizzare per la creazione di file è:

NomeDocumento_vX.Y.Z.pdf

3.1.2.6 Strumenti

3.1.2.6.1 \LaTeX

Per redigere la documentazione si è scelto di utilizzare il linguaggio di markup \LaTeX , in quanto:

- permette di produrre documenti di alta qualità rispetto ai *word processor*_g tradizionali;
- rende possibile creare dei template comuni per i documenti;
- permette una separazione tra contenuto e formattazione;
- è estendibile ed altamente personalizzabile tramite l'utilizzo di pacchetti specifici;
- è multiplatforma, essendo un file sorgente di \LaTeX una semplice codifica *ASCII*_g;

Come editor di file \LaTeX è consigliato l'uso di *TeXstudio*, anch'esso multiplatforma.

3.1.2.6.2 Correttore ortografico

In *TeXstudio* è presente un correttore ortografico automatico che permette la correzione in tempo reale. I file da utilizzare sono presenti nella cartella condivisa in Google Drive all'indirizzo [/dizionari](#).

3.1.2.6.3 Script

Su Github è presente una cartella [/script](#) dove sono presenti degli script utilizzabili dal gruppo per semplificare alcune operazioni.

3.2 Processo di verifica

3.2.1 Attività

3.2.1.1 Verifica di un ticket

Ogni qual volta un ticket sarà contrassegnato come **Completato** il Responsabile procederà nel seguente modo:

- crea un ticket di verifica;
- lo assegna ad un **verificatore**;
- non appena il ticket viene contrassegnato come **Completato** imposta il tag del ticket oggetto di verifica a **Verificato**.

3.2.2 Procedure

3.2.2.1 Gestione delle anomalie

Il **validatore** qualora riscontrasse delle anomalie procederà secondo la seguente procedura:

- crea un nuovo subtask del ticket di verifica per ogni anomalia riscontrata;
- assegna un titolo breve e preciso ad ogni subtask;
- se necessario aggiunge un commento che descriva l'anomalia riscontrata;
- assegna i subtasks al redattore del documento.

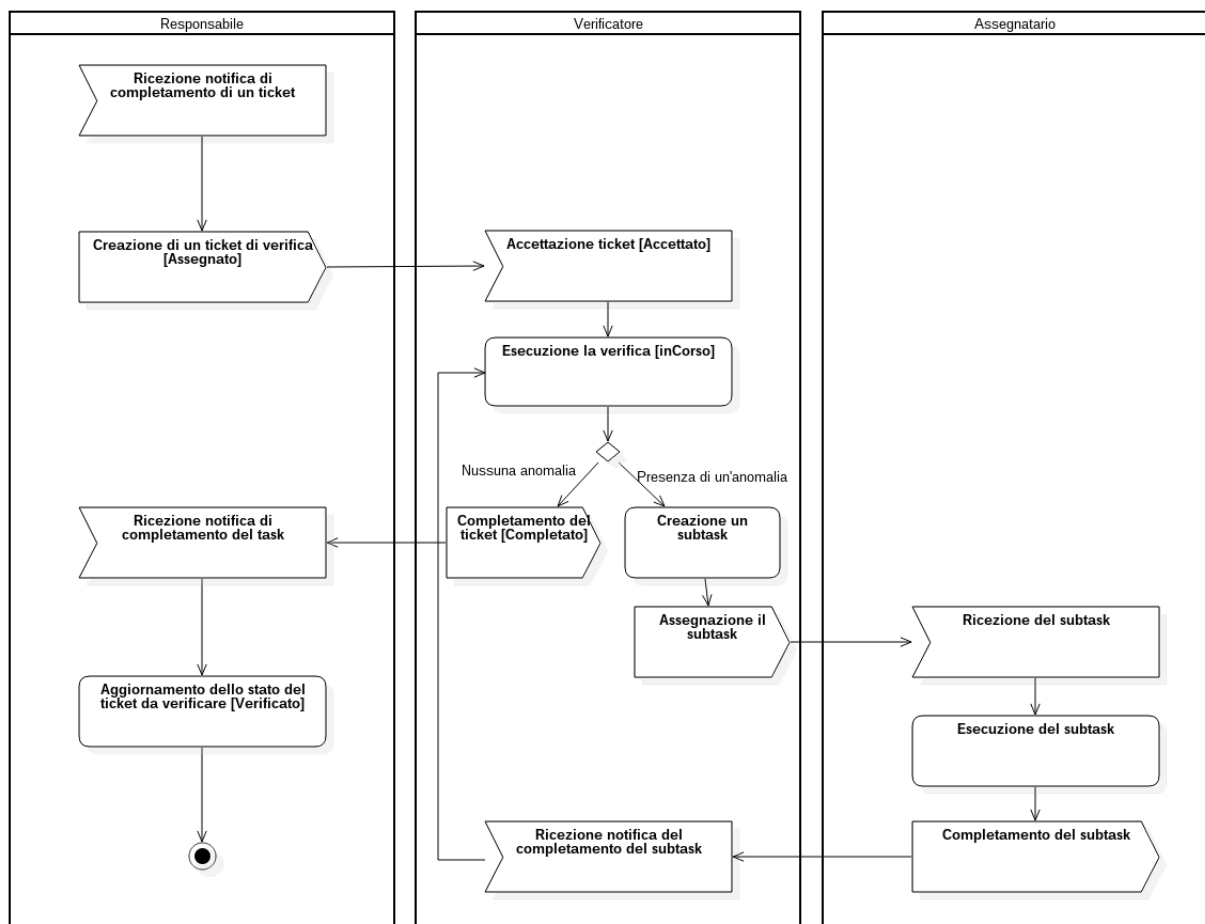


Figura 1: Ticket di verifica

4 Processi organizzativi

4.1 Processo di gestione

4.1.1 Attività

4.1.1.1 Comunicazioni

4.1.1.1.1 Comunicazioni interne

Per le comunicazioni interne si è deciso di utilizzare *Slack_g*. Sono stati creati diversi canali per suddividere il lavoro. I canali sono:

- *#general*: in questo canale va utilizzato per le comunicazioni di carattere generale e per le convocazioni delle riunioni;
- *#random*: in caso di piccoli dubbi o questioni di scarsa rilevanza, si consiglia l'utilizzo di questo canale;
- *#tools*: questo canale è stato creato per domande relative a dubbi e/o problemi sugli strumenti utilizzati per lo sviluppo;
- *#git*: su questo canale è stato collegato un *bot_g* che si occupa di notificare tutti i commit effettuati da ogni singolo componente del gruppo;
- *#nome_del_documento*: per ogni documento è stato creato un canale dove è possibile discutere di tutte le scelte da prendere per quel singolo documento.

In caso di necessità, sarà possibile aggiungere nuovi canali nelle fasi successive dello sviluppo. Inoltre è possibile utilizzare il gruppo creato su *Telegram_g* per comunicazioni interne non inerenti al progetto. Per le videoconferenze invece verrà utilizzato *Google Hangouts_g*.

4.1.1.1.2 Comunicazioni esterne

Per le comunicazioni esterne è stata creata la casella di posta elettronica:

beaconstrips.swe@gmail.com

Questo indirizzo deve essere l'unico usato per comunicare con le componenti esterne al team ed è controllato unicamente dal Responsabile. Il Responsabile è tenuto poi ad informare i membri del team riguardo le comunicazioni avvenute con l'esterno.

4.1.1.2 Riunioni

4.1.1.2.1 Riunioni interne

Il gruppo si ritroverà ad organizzare una riunione almeno ogni dieci giorni. Il Responsabile si occuperà di convocare su Slack le riunioni generali cui tutti i membri del team sono convocati, avvisando i componenti con almeno due giorni di preavviso. In caso di necessità un componente del team può richiedere la convocazione di una riunione: tale richiesta deve essere inoltrata al Responsabile, che deciderà se accettarla o respingerla. Inoltre sono possibili riunioni tra specifici membri: in questo caso sono tenuti ad informare il resto del team tramite un verbale, nel caso siano state prese decisioni rilevanti.

4.1.1.2.2 Riunioni esterne

Il Responsabile si occuperà di concordare con il proponente o con i committenti le riunioni esterne al gruppo. È gradita la presenza di tutti i componenti del gruppo ad ogni incontro.

Prima dell'inizio di ogni riunione verrà scelto un componente del gruppo che svolgerà la funzione di segretario, il quale annoterà ogni argomento trattato e si occuperà di redigere un verbale che andrà poi inviato agli altri componenti del team.

4.1.1.3 Ticket

I ticket avranno la seguente struttura:

- **Titolo:** deve rappresentare in modo chiaro e conciso il task;
- **Scadenza:** data entro cui l'assegnatario deve concludere il task;
- **Descrizione:** se necessario, deve chiarire con una breve descrizione il task;
- **Assegnatario:** colui che dovrà completare il task;
- **Tag di stato:** deve contenere il giusto tag che rappresenta lo stato attuale del task;
- **Subtasks:** il task può essere suddiviso in task più piccoli.

4.1.1.3.1 Lista dei tag

Ogni ticket dovrà sempre contenere un tag che rappresenti lo stato attuale di avanzamento:

- **Assegnato:** il ticket è stato assegnato ad un membro del team;
- **Accettato:** l'assegnatario ha preso in carico il ticket;
- **Rifiutato:** l'assegnatario ha rifiutato il ticket ed ha inserito la motivazione in un commento;
- **InCorso:** l'assegnatario sta svolgendo il task;
- **InSospeso:** l'assegnatario specifica della motivazione per cui ha messo in attesa il ticket;
- **Completato:** l'assegnatario ha concluso il task;
- **Verificato:** il ticket è stato verificato.

4.1.2 Procedure

4.1.2.1 Creazione di un ticket

La creazione e l'assegnazione dei ticket sarà delegata al Responsabile che seguirà la seguente procedura:

- assegna il **titolo** del ticket;
- assegna la data di **scadenza** del ticket;
- se necessario descrive il task più approfonditamente tramite la **descrizione**;
- assegna il ticket ad un **assegnatario**;
- imposta il **tag di stato** ad **Assegnato**.



Figura 2: Creazione di un ticket

4.1.2.2 Aggiornamento di un ticket

Ogni membro del gruppo avrà una lista di ticket a lui assegnati, su di essi dovrà modificare il tag di stato ogni qual volta uno di questi cambi di stato come descritto dalla sezione 4.1.1.3.1. Nel caso in cui il ticket venga **rifiutato** o venga messo **in sospeso** è necessario scrivere una motivazione attraverso un commento.

4.1.3 Norme

4.1.3.1 Repository

Per la gestione condivisa e per il controllo delle versioni dei file il *team_g* ha deciso di utilizzare una repository fornita dal servizio web *GitHub_g*.

4.1.3.1.1 Struttura del repository

La repository si divide in due cartelle principali:

- **Codice:** contenente il codice del progetto, con struttura ancora da definire;
- **Documenti:** contenente:
 1. **template:** con al suo interno i file **.sty** e una cartella **img** contenente le immagini (vedi sezione 3.1.2.1);
 2. una cartella per ogni **revisione di avanzamento**, il cui nome è definito dal numero e dalla sigla della revisione, separata da un trattino. Ciascuna di esse è suddivisa in:
 - (a) **interni:** contenente una cartella per ogni documento interno, con nome, in notazione *CamelCase_g*, uguale a quello del documento(senza numero di versione);
 - (b) **esterni:** contenente una cartella per ogni documento esterno, con nome, in notazione *CamelCase_g*, uguale a quello del documento(senza numero di versione);

4.1.3.1.2 Tipi di file e .gitignore

All'interno delle cartelle dei documenti saranno presenti solamente i file **.tex**, i restanti file ausiliari prodotti da *L^AT_EX* sono stati aggiunti a *.gitignore* e quindi vengono ignorati e resi invisibili da *Git_g*.

4.1.3.1.3 Norme sulla commit

Per rendere effettive le modifiche applicate ai file della repository locale è necessario eseguire una **commit**. I membri del team sono tenuti ad effettuare una commit ogni qual volta venga aggiunta, rimossa o aggiornata una funzionalità oppure una sezione, aggiungendo i file interessati tramite il comando:

```
git add nome.del.file.est
```

ed eseguendo la commit tramite il comando:

```
git commit -m 'messaggio'
```

con **messaggio** che segua le regole della sezione 3.2.1. Infine per rendere effettive le modifiche nel repository in remoto eseguire:

```
git push
```

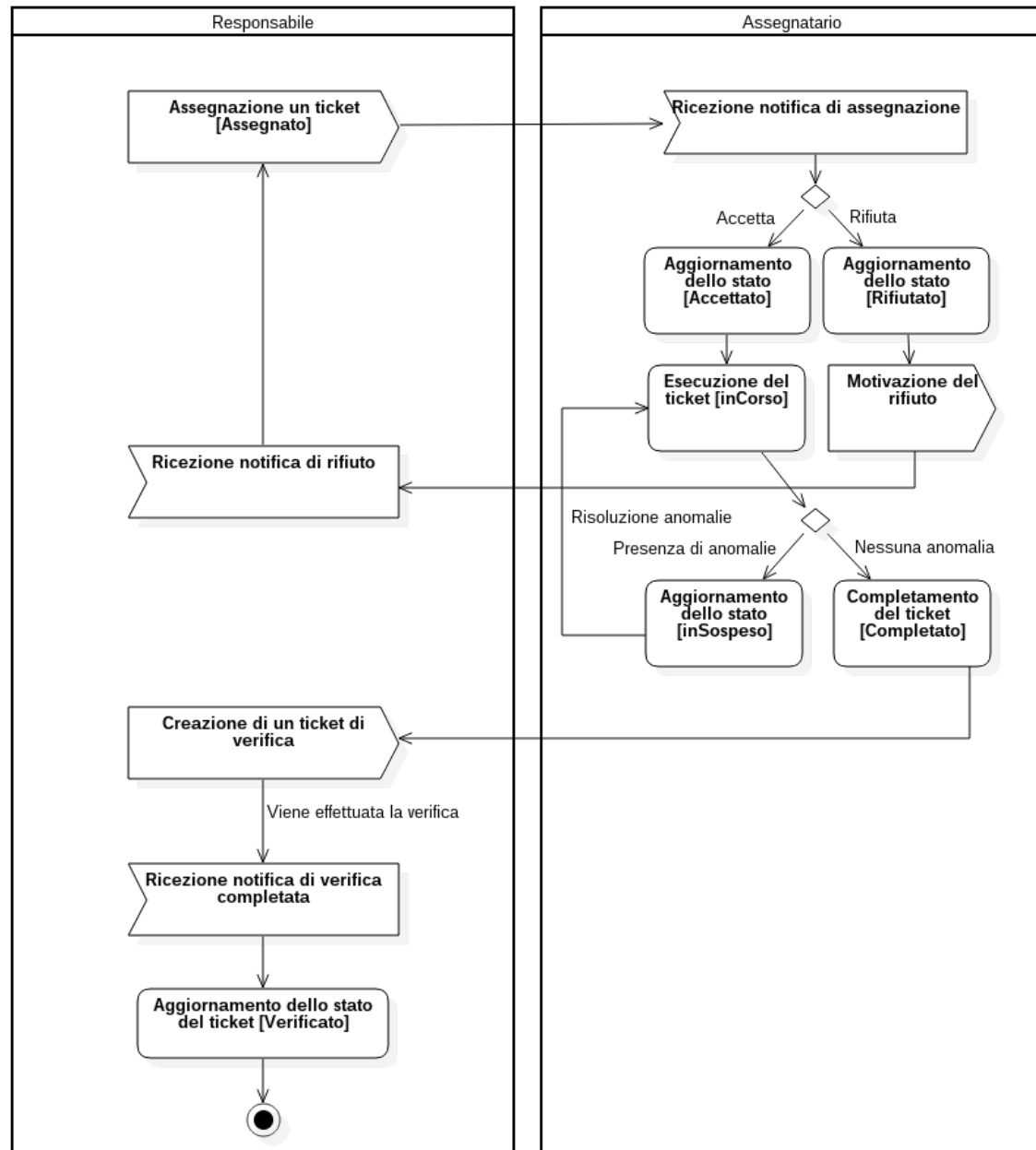


Figura 3: Ciclo di vita di un ticket

4.1.3.1.4 Messaggi delle commit

Ogni qual volta venga eseguita una **commit** è obbligatorio inserire un breve messaggio con la spiegazione delle modifiche apportate; il messaggio deve iniziare con:

- **Add:** se vengono aggiunte funzionalità o sezioni;
- **Remove:** se vengono tolte funzionalità o sezioni;
- **Update:** se vengono aggiornate funzionalità o sezioni;
- **Fix:** se viene sistemato un bug noto;

deve seguire una descrizione sintetica in lingua italiana delle modifiche apportate.

4.1.4 Strumenti

4.1.4.1 Sistema operativo

Non ci sono restrizioni riguardo l'utilizzo di un sistema operativo: ogni membro del team può utilizzare quindi il sistema operativo che preferisce; in ogni caso, è tenuto a rendere il materiale prodotto **completamente** compatibile su tutte le piattaforme.

4.1.4.2 Condivisione

Il servizio da utilizzare per i documenti informali è *Google Drive_g*. All'interno di questo repository vanno caricati documenti che:

- non necessitano di controllo di versione;
- necessitano di essere modificati contemporaneamente da più persone; infatti Google Drive permette la modifica di un documento in contemporanea: in questo modo non ci saranno conflitti.

È possibile anche condividere link o manuali utili per la formazione del team. Si può installare Google Drive sul proprio PC: i file così potranno essere disponibili anche offline e sincronizzare il contenuto della cartella ad ogni aggiornamento.

4.1.4.3 Gestione

- **Ticket:** per la gestione dei ticket è stato scelto di utilizzare *Asana_g*; per la gestione dei ticket, si rimanda alla sezione
- **Requisiti e casi d'uso:** per il tracciamento dei requisiti si utilizzerà *Trender_g*. Per quanto riguarda la rappresentazione dei casi d'uso l'editor consigliato è *Star UML_g*;
- **Scadenze:** *Google Calenar_g* verrà utilizzato per segnare le scadenze del gruppo; si riceverà una notifica su Slack trenta minuti prima dell'evento;
- **Progetto:** per la realizzazione dei *grafici di Gantt_g* è stato impiegato il programma *Gantt-Project_g*.