



Norme di Progetto

Informazioni sul documento

Versione	v1.0.0
Data di Creazione	2016-12-03
Data ultima modifica	2016-12-23
Stato	Approvato
Redazione	Giovanni Damo Giovanni Prete
Verifica	Leonardo Brutesco
Approvazione	Giulia Petenazzi
Uso	Interno
Lista di distribuzione	Professor Tullio Vardanega Professor Riccardo Cardin Zephyrus
Email di riferimento	zephyrus.swe@gmail.com

Registro delle modifiche

Versione	Data	Autore	Ruolo	Descrizione
1.0.0	2016-12-23	Giulia Petenazzi	<i>Responsabile</i>	Approvazione
0.2.0	2016-12-20	Leonardo Brutesco	<i>Verificatore</i>	Verifica documento
0.1.1	2016-12-20	Giovanni Prete	<i>Analista</i>	Correzione errori
0.1.0	2016-12-17	Leonardo Brutesco	<i>Verificatore</i>	Verifica documento
0.0.10	2016-12-17	Giovanni Prete	<i>Analista</i>	Correzioni e aggiornamento diagrammi
0.0.9	2016-12-16	Giovanni Damo	<i>Analista</i>	Correzioni sezione processi primari
0.0.8	2016-12-13	Giovanni Prete	<i>Analista</i>	Correzioni sezione processi di organizzativi
0.0.7	2016-12-09	Giovanni Damo	<i>Analista</i>	Correzioni sezione processi di supporto
0.0.6	2016-12-07	Giovanni Prete	<i>Analista</i>	Aggiunte descrizioni attività e processi
0.0.5	2016-12-07	Giovanni Damo	<i>Analista</i>	Terminata sezione processi di supporto
0.0.4	2016-12-05	Giovanni Damo	<i>Analista</i>	Terminata sezione processi organizzativi
0.0.3	2016-12-05	Giovanni Prete	<i>Analista</i>	Terminata sezione processi primari
0.0.2	2016-12-04	Giovanni Prete	<i>Analista</i>	Terminata introduzione
0.0.1	2016-12-03	Giovanni Prete	<i>Analista</i>	Creazione template e indice

Indice

1	Introduzione	1
1.1	Scopo del documento	1
1.2	Scopo del prodotto	1
1.3	Glossario	1
1.4	Riferimenti	2
1.4.1	Riferimenti normativi	2
1.4.2	Riferimenti informativi	2
2	Processi primari	3
2.1	Fornitura	3
2.1.1	Scopo	3
2.1.2	Studio di fattibilità	3
2.1.2.1	Scopo	3
2.1.2.2	Discussione e scelta del capitolato	3
2.1.2.3	Struttura studio di fattibilità	3
2.1.3	Contrattazione	3
2.1.3.1	Scopo	3
2.1.3.2	Preparazione della proposta	4
2.2	Sviluppo	4
2.2.1	Scopo	4
2.2.2	Analisi dei requisiti	4
2.2.2.1	Scopo	4
2.2.2.2	Classificazione dei casi d'uso	4
2.2.2.3	Classificazione dei requisiti	5
2.2.2.4	Diagrammi UML	6
2.2.3	Progettazione ad alto livello	6
2.2.3.1	Scopo	6
2.2.3.2	Specifica tecnica	6
2.2.3.2.1	Diagrammi UML	6
2.2.3.2.2	Design pattern	6
2.2.3.2.3	Tracciamento delle componenti	7
2.2.3.2.4	Test di integrazione	7
2.2.4	Progettazione a basso livello	7
2.2.4.1	Scopo	7
2.2.4.2	Definizione di prodotto	7
2.2.4.2.1	Diagrammi UML	7
2.2.4.2.2	Definizione delle classi	7
2.2.4.2.3	Tracciamento delle classi	7
2.2.4.2.4	Test di unità	7
2.2.5	Codifica e test	7
2.2.5.1	Scopo	7
2.2.5.2	Stile di codifica	8
2.2.5.3	Ricorsione	8
2.2.5.4	Variabili globali	8
2.2.6	Strumenti	8
2.2.6.1	Trender	8
2.2.6.2	Astah	9

3	Processi di supporto	10
3.1	Documentazione	10
3.1.1	Scopo	10
3.1.2	Template	10
3.1.3	Struttura dei documenti	10
3.1.3.1	Prima pagina	10
3.1.3.2	Registro delle modifiche	10
3.1.3.3	Indice	11
3.1.3.4	Note a piè di pagina	11
3.1.3.5	Contenuto principale	11
3.1.4	Norme tipografiche	11
3.1.4.1	Glossario	11
3.1.4.2	Stile del testo	11
3.1.4.3	Titoli	12
3.1.4.4	Elenchi puntati	12
3.1.4.5	Formati	13
3.1.4.6	Nomi propri	13
3.1.5	Componenti grafiche	14
3.1.5.1	Tabelle	14
3.1.5.2	Immagini	14
3.1.6	Classificazione dei documenti	14
3.1.6.1	Documenti informali	14
3.1.6.2	Documenti formali	14
3.1.6.3	Verbalì	14
3.1.6.4	Versionamento	15
3.1.7	Procedure	16
3.1.7.1	Creazione di un documento	16
3.1.7.2	Ciclo di vita dei documenti	16
3.1.8	Strumenti	17
3.1.8.1	Latex	17
3.1.8.2	TeXstudio	17
3.2	Verifica	17
3.2.1	Scopo	17
3.2.2	Procedure	17
3.2.2.1	Analisi	17
3.2.2.1.1	Analisi statica	17
3.2.2.1.2	Analisi dinamica	18
3.2.2.2	Task di verifica	18
3.2.2.3	Gestione delle anomalie	18
3.2.3	Strumenti	19
3.2.3.1	Verifica ortografica	19
3.2.3.2	Indice leggibilità	19
3.2.3.3	Validazione web	19
3.3	Validazione	20
3.3.1	Scopo	20
4	Processi organizzativi	21
4.1	Gestione processi	21
4.1.1	Scopo	21

4.1.2	Comunicazioni	21
4.1.2.1	Interne	21
4.1.2.2	Esterne	21
4.1.3	Incontri	21
4.1.3.1	Interni	21
4.1.3.2	Esteri	21
4.1.4	Ruoli di progetto	21
4.1.4.1	Responsabile di progetto	22
4.1.4.2	Amministratore di progetto	22
4.1.4.3	Analista	22
4.1.4.4	Progettista	22
4.1.4.5	Programmatore	23
4.1.4.6	Verificatore	23
4.1.5	Procedure	23
4.1.5.1	Organizzazione incontri interni	23
4.1.5.2	Organizzazione incontri esterni	23
4.1.5.3	Gestione dei ticket	24
4.1.5.3.1	Creazione di un ticket	24
4.1.5.3.2	Modifica di un ticket	24
4.1.5.4	Gestione delle milestone	26
4.1.5.4.1	Creazione milestone	26
4.1.5.4.2	Modifica milestone	27
4.1.6	Strumenti	27
4.1.6.1	Slack	27
4.1.6.2	Teamwork	28
4.1.6.3	Condivisione file	28
4.2	Gestione delle infrastrutture	29
4.2.1	Scopo	29
4.2.2	Struttura dei repository	29
4.2.3	Nomi dei file	30
4.2.4	Messaggi di commit	30
4.2.5	Procedure	30
4.2.5.1	Installazione e configurazione di Git	30
4.2.5.2	Invio di commit	31
4.2.5.3	Comandi utili Git	31
4.2.6	Strumenti	31
4.2.6.1	Git	31
4.2.6.2	GitHub	32
4.2.6.3	Sistemi operativi	32
4.3	Apprendimento	32
4.3.1	Scopo	32

A Lista di controllo

33

1 Introduzione

1.1 Scopo del documento

Questo documento definisce le norme che i membri del *gruppo_G* Zephyrus si impegnano a rispettare per un corretto svolgimento del progetto *DeGeOP*. Ogni componente del *gruppo_G* è tenuto a leggere e rispettare quanto scritto al fine di:

- garantire uniformità nel materiale prodotto;
- favorire la cooperazione tra i membri del *gruppo_G*;
- raggiungere il miglior rapporto tra efficacia ed efficienza.

Il presente documento definisce le norme inerenti a:

- comunicazioni interne ed esterne al *gruppo_G*;
- stesura dei documenti e convenzioni tipografiche;
- stesura del codice;
- modalità di lavoro durante le *fasig*;
- organizzazione dell'ambiente di lavoro e di sviluppo.

Ogni membro del *gruppo_G* sarà informato di eventuali modifica alle norme esistenti o aggiunta di nuove norme.

1.2 Scopo del prodotto

Lo scopo del prodotto consiste nella creazione di un'interfaccia web contenente una mappa geografica su cui potranno essere rappresentati:

- il processo produttivo aziendale;
- gli scenari di danno;
- i risultati dell'analisi dei rischi.

Il prodotto verrà utilizzato da agenti assicuratori per l'inserimento delle informazioni utili allo svolgimento dell'analisi dei rischi dell'assicurando.

L'interfaccia dovrà essere in grado di connettersi ai sistemi preesistenti di *RiskApp* per la memorizzazione e gestione dei dati inseriti.

1.3 Glossario

Allo scopo di rendere più semplice e chiara la comprensione dei documenti viene allegato il *Glossario v1.0.0*, nel quale verranno raccolte le spiegazioni di terminologia tecnica o ambigua, abbreviazioni ed acronimi. Per evidenziare un termine presente in tale documento, esso verrà marcato con il pedice *G*.

Tutti i termini del glossario evidenziati sono link ipertestuali al glossario stesso; affinché funzionino correttamente è necessario che la posizione delle directory e dei file forniti non venga alterata.

1.4 Riferimenti

1.4.1 Riferimenti normativi

- standard **ISO_G-8601**:
https://en.wikipedia.org/wiki/ISO_8601;
- standard **ISO_G/IEC_G 12207**:
https://en.wikipedia.org/wiki/ISO/IEC_12207;
- specifica **UML_G 2.0**:
<http://www.omg.org/spec/UML/2.0/>.

1.4.2 Riferimenti informativi

- standard AS/NZS **ISO_G/IEC_G 12207:1997**:
http://www.math.unipd.it/~tullio/IS-1/2009/Approfondimenti/ISO_12207-1995.pdf;
- **capitolato_G**:
<http://www.math.unipd.it/~tulXlio/IS-1/2016/Progetto/C3.pdf>;
- utilizzo di **Teamwork_G**:
<http://support.teamwork.com/projects/start/getting-started>;
- utilizzo di **Git_G**:
<https://git-scm.com/doc>;
- utilizzo di **GitHub_G**:
<https://guides.github.com/>;
- utilizzo di **Astah_G**:
<http://astah.net/tutorials>;
- utilizzo di **Slack_G**:
<https://get.slack.help/hc/en-us>.

2 Processi primari

2.1 Fornitura

2.1.1 Scopo

Lo scopo del processo di fornitura è di determinare le procedure e le risorse necessarie allo svolgimento del progetto. Le attività di cui si compone sono:

- studio di fattibilità;
- contrattazione.

La corretta implementazione del processo deve:

- decidere il progetto da svolgere;
- fissare gli obiettivi per la contrattazione.

2.1.2 Studio di fattibilità

2.1.2.1 Scopo

Individuare gli aspetti fondamentali dei progetti proposti, tramite lo studio dei capitolati ed il confronto tra i membri del *gruppo_G*.

2.1.2.2 Discussione e scelta del capitolato

Il *Responsabile di progetto* ha il compito di organizzare gli incontri del *gruppo_G* necessari ad analizzare i capitolati proposti.

2.1.2.3 Struttura studio di fattibilità

Il documento creato dagli *Analisti* per ogni *capitolato_G* deve rispettare i seguenti punti:

- **descrizione:** descrizione del prodotto richiesto dal *capitolato_G*;
- **dominio applicativo:** ambito di utilizzo del prodotto;
- **dominio tecnologico:** tecnologie richieste per lo sviluppo del progetto;
- **criticità:** individuazione di punti critici e possibili problematiche che potrebbero sorgere durante lo svolgimento del progetto;
- **valutazione finale:** considerazioni finali sulla scelta di accettare o meno il *capitolato_G* preso in esame.

2.1.3 Contrattazione

2.1.3.1 Scopo

Presentare una proposta in risposta al *capitolato_G* del proponente.

2.1.3.2 Preparazione della proposta

Il *gruppo_G* deve redigere e consegnare i seguenti documenti:

- *Norme di progetto;*
- *Studio di fattibilità;*
- *Analisi dei requisiti;*
- *Piano di progetto;*
- *Piano di qualifica.*

Verrà inoltre fornita in allegato la *Lettera di presentazione* del *gruppo_G*. Si veda la sezione 3.1.6 per maggiori informazioni sulla gestione dei documenti.

2.2 Sviluppo

2.2.1 Scopo

Il processo di sviluppo contiene le attività necessarie a produrre il prodotto software richiesto. In accordo con lo standard *ISO_G/IEC_G* (1.4.2) preso come riferimento le attività che lo compongono sono:

- analisi dei requisiti;
- progettazione;
- codifica.

La corretta implementazione del processo deve:

- fissare gli obiettivi di sviluppo;
- fissare i vincoli tecnologici;
- realizzare un prodotto finale che soddisfi i test di accettazione e che sia conforme alle richieste del proponente.

2.2.2 Analisi dei requisiti

2.2.2.1 Scopo

Individuare i requisiti del progetto tramite lo studio del *capitolato_G* ed incontri con il proponente. Individuare i test di sistema. Il risultato deve essere presentato nel documento formale *Analisi dei requisiti* che deve contenere la lista dei casi d'uso e dei requisiti.

2.2.2.2 Classificazione dei casi d'uso

I casi d'uso individuati devono essere classificati secondo la seguente notazione:

UC[Codice padre].[Codice identificativo]

dove:

- **codice padre:** indica il codice numerico in forma gerarchica del caso d'uso da cui deriva, viene omesso se non identificabile;

- **codice identificativo:** indica il codice numerico del caso d'uso.

Per ogni caso d'uso bisogna indicare:

- **titolo:** titolo riassuntivo dell'operazione che il caso d'uso modella;
- **attori:** elenco attori coinvolti;
- **descrizione:** concisa e meno ambigua possibile;
- **pre-condizione:** condizioni sempre vere riferite allo stato del sistema che abilitano lo svolgimento del caso d'uso;
- **post-condizione:** condizioni sempre vere riferite allo stato del sistema dopo lo svolgimento del caso d'uso;
- **scenario principali:** ordine con cui vengono eseguiti i casi d'uso figli;
- **scenari alternativi:** possibili scenari alternativi del caso d'uso;
- **estensioni:** spiegazione di tutte le estensioni se presenti;
- **inclusioni:** spiegazione di tutte le inclusioni se presenti;
- **generalizzazioni:** spiegazione di tutte le generalizzazioni se presenti.

2.2.2.3 Classificazione dei requisiti

I requisiti individuati devono essere classificati secondo la seguente notazione:

R[Importanza][Tipologia][Codice]

dove:

- **importanza:** può assumere questi valori:
 - **O:** indica un requisito obbligatorio;
 - **D:** indica un requisito desiderabile;
 - **F:** indica un requisito opzionale (facoltativo).
- **tipologia:** può assumere questi valori:
 - **F:** indica un requisito funzionale;
 - **Q:** indica un requisito di qualità;
 - **P:** indica un requisito prestazionale;
 - **V:** indica un requisito di vincolo.
- **codice:** codice numerico che identifica il requisito, deve essere univoco ed indicato in forma gerarchica, da sinistra a destra, nella notazione X.Y.Z.

Per ogni requisito bisogna inoltre indicare:

- **descrizione:** concisa e meno ambigua possibile;
- **fonte:** l'origine dei requisiti deve essere una delle seguenti:

- **capitolato_G**: derivato direttamente dal testo del **capitolato_G**;
- **interno**: deriva da discussioni interne al **gruppo_G**;
- **verbale**: deriva da un verbale steso in seguito ad un incontro con il proponente. Deve essere indicato il nome del verbale a cui si riferisce;
- **casi d'uso**: deriva da uno o più casi d'uso. Deve essere indicato il codice identificativo dei casi d'uso a cui si riferisce.

2.2.2.4 Diagrammi UML

I diagrammi **UML_G** devono essere realizzati seguendo lo standard **UML_G** versione 2.

Per facilitare la lettura dei diagrammi si devono seguire le seguenti convenzioni generali:

- gli elementi devono essere il più possibile omogenei tra loro per dimensione;
- gli elementi devono essere allineati tra loro, sia orizzontalmente che verticalmente, quando possibile;
- i margini di spazio tra gli elementi di un **gruppo_G** devono rimanere invariati in gruppi analoghi se si hanno le stesse tipologie di elementi;
- i collegamenti in uscita da un singolo elemento devono essere ad angolo retto invece che diretti.

2.2.3 Progettazione ad alto livello

2.2.3.1 Scopo

Definire la struttura di alto livello del software e identificare le sue componenti. Definire le interfacce esterne ed interne. Individuare i test di integrazione. Il risultato deve essere presentato nel documento formale *Specifica tecnica*.

2.2.3.2 Specifica tecnica

2.2.3.2.1 Diagrammi UML

Devono essere forniti i seguenti diagrammi:

- diagrammi di classe;
- diagrammi dei **package_G**;
- diagrammi di attività;
- diagrammi di sequenza.

2.2.3.2.2 Design pattern

Devono essere fornita una descrizione dei **design pattern_G** utilizzati per la realizzazione dell'architettura. Ogni **design pattern_G** deve avere una descrizione ed un diagramma che ne esponga la struttura.

2.2.3.2.3 Tracciamento delle componenti

Tutti i requisiti devono essere riferiti al componente che li soddisfa per poter verificare che ogni requisito sia soddisfatto. Si veda la sezione [2.2.6.1](#) in cui viene descritto lo strumento utilizzato per il tracciamento.

2.2.3.2.4 Test di integrazione

Devono essere definite le classi di verifica necessarie a garantire che tutte le componenti del sistema funzionino correttamente.

2.2.4 Progettazione a basso livello

2.2.4.1 Scopo

Definire la struttura di tutte le componenti e suddividerle in unità che possano essere realizzate, compilate e testate singolarmente. Individuare i test delle unità. Il risultato deve essere presentato nel documento formale *Definizione di prodotto*.

2.2.4.2 Definizione di prodotto

2.2.4.2.1 Diagrammi UML

Devono essere forniti i seguenti diagrammi:

- diagrammi di classe;
- diagrammi di attività;
- diagrammi di sequenza.

2.2.4.2.2 Definizione delle classi

Deve essere fornita una descrizione per ogni classe progettata che ne spieghi lo scopo e che definisca le funzionalità.

2.2.4.2.3 Tracciamento delle classi

Tutti i requisiti devono essere tracciati alle classi associate per poter verificare che ogni classe soddisfi almeno un requisito. Si veda la sezione [2.2.6.1](#) in cui viene descritto lo strumento utilizzato per il tracciamento.

2.2.4.2.4 Test di unità

Devono essere definiti dei test di unità necessari a garantire che tutte le componenti del sistema funzionino correttamente.

2.2.5 Codifica e test

2.2.5.1 Scopo

Sviluppare le unità ed i test individuati durante la progettazione. Il risultato finale deve essere il codice sorgente del prodotto da realizzare e dei test necessari.

2.2.5.2 Stile di codifica

Al fine di produrre codice uniforme, leggibile e manutenibile è richiesto che vengano rispettate le seguenti convenzioni:

- i nomi utilizzati devono essere chiari, descrittivi rispetto alla loro funzione e in inglese;
- evitare nomi troppo simili tra loro che possano creare difficoltà nella comprensione del codice;
- deve essere presente almeno un breve commento descrittivo per ogni classe e metodo;
- i commenti devono essere scritti in lingua italiana senza utilizzare abbreviazioni o altre ambiguità;
- le modifiche al codice devono sempre riflettersi sui relativi commenti;
- evitare commenti superflui, inappropriati o scurrili;
- ogni file deve presentare un'intestazione con le seguenti informazioni:
 - percorso e nome del file;
 - nome e cognome dell'autore;
 - data di creazione;
 - breve descrizione del contenuto del file.

2.2.5.3 Ricorsione

La ricorsione va sempre evitata se possibile. Per ogni funzione ricorsiva è necessario fornire una prova di terminazione nei commenti.

2.2.5.4 Variabili globali

L'uso di variabili globali va sempre evitato se possibile.

2.2.6 Strumenti

2.2.6.1 Trender

Il *gruppoG* utilizza l'applicazione *TrenderG* per gestire i dati ricavati dall'analisi dei requisiti. *TrenderG* è stato sviluppato dal *gruppoG* InfiniTech per il progetto dell'anno 2014/15, utilizza un *databaseG MySQLG* per la persistenza dei dati ed è disponibile al seguente indirizzo:

<https://github.com/campagna91/Trender>.

Le funzionalità offerte da *TrenderG* sono:

- tracciamento dei requisiti;
- tracciamento dei casi d'uso;
- tracciamento dei verbali;
- tracciamento degli attori presenti nel sistema;
- tracciamento dei packages;

- tracciamento delle classi;
- tracciamento dei test;
- tracciamento delle voci del glossario;
- possibilità di creare il codice \LaTeX di quanto archiviato nei punti precedenti.

Lo spazio di lavoro dedicato al *gruppoG* si trova al seguente indirizzo:

<http://zephyrusar.altervista.org/trender/>

2.2.6.2 Astah

AstahG è un'applicazione per la creazione di diagrammi *UMLG* ed è utilizzata nel corso delle attività di analisi e progettazione. I diagrammi che verranno realizzati sono:

- diagrammi di sequenza;
- di attività;
- dei casi d'uso;
- delle classi.

Le principali motivazioni che hanno portato il *gruppoG* alla scelta di questo strumento sono:

- già utilizzato da parte del docente;
- versione professional gratuita con la licenza studente;
- *cross-platformG*.

Indirizzi per il download del programma e della licenza studente:

<http://astah.net/download>
<http://astah.net/student-license-request>

3 Processi di supporto

3.1 Documentazione

3.1.1 Scopo

Lo scopo del processo di documentazione è di redigere e mantenere la documentazione durante l'intero *ciclo di vita_G* del software. La corretta implementazione del processo deve:

- dare una chiara visione dei documenti che devono essere prodotti durante il *ciclo di vita_G* del software;
- fornire le norme necessarie alla stesura dei documenti;
- produrre documenti formali coerenti.

3.1.2 Template

È stato creato un template \LaTeX per garantire che tutti i documenti creati dal *gruppo_G* abbiano la stessa struttura grafica e lo stesso stile di formattazione. Ogni membro del *gruppo_G* deve creare i documenti richiesti utilizzando tale template e deve ridurre al minimo indispensabile eventuali variazioni personali nella formattazione.

3.1.3 Struttura dei documenti

3.1.3.1 Prima pagina

La prima pagina di ogni documento deve presentare la seguente struttura:

- logo del *gruppo_G*;
- titolo del documento;
- informazioni del documento:
 - versione;
 - data di creazione;
 - data di ultima modifica;
 - stato del documento;
 - redattori del documento;
 - verificatori;
 - responsabile approvazione;
 - lista di distribuzione;
 - email del *gruppo_G*.

3.1.3.2 Registro delle modifiche

La seconda pagina di ogni documento formale deve contenere una tabella con la lista delle modifiche apportate al documento. Ogni riga della tabella deve essere compilata per intero con le seguenti informazioni:

- versione del documento successiva alla modifica;

- data in cui è stata eseguita la modifica;
- autore della modifica;
- ruolo dell'autore;
- descrizione concisa della modifica apportata.

3.1.3.3 Indice

Nella pagina successiva alla fine del registro delle modifiche ogni documento formale deve contenere l'indice dei suoi contenuti; tale indice deve permettere la lettura *ipertestuale*_G del documento. L'indice deve essere numerato a partire da 1, ogni sottosezione riparte da 1 e aggiunge il proprio indice a quello del padre separandolo con un punto. Eventuali appendici invece di essere numerate saranno indicate con lettere maiuscole a partire da A seguendo l'ordine alfabetico internazionale.

3.1.3.4 Note a piè di pagina

Eventuali note vanno indicate in basso a sinistra nella pagina in cui compaiono con il loro numero identificativo e la loro descrizione.

3.1.3.5 Contenuto principale

Tutte le pagine successive all'indice del documento devono contenere un'intestazione e un piè di pagina. L'intestazione deve contenere:

- nome della sezione allineato a sinistra;
- nome del documento allineato a destra.

Il piè di pagina deve contenere:

- nome del *gruppo*_G e del progetto allineati a sinistra;
- pagina corrente rispetto alle pagine totali del documento allineate a destra.

3.1.4 Norme tipografiche

3.1.4.1 Glossario

Ogni parola contenuta nel *Glossario* deve essere scritta in corsivo e contrassegnata da una "G" a pedice come da esempio:

*termine*_G

3.1.4.2 Stile del testo

Per una facilitare la stesura del documento, migliorarne correttezza e leggibilità, TexStudio mette a disposizione dei tool per il controllo grammaticale. Ogni membro del *gruppo*_G deve controllare nelle impostazioni del proprio strumento che siano attivati:

- **individua ripetizioni:** durante la scrittura del documento, se una parola viene ripetuta troppo viene sottolineata di verde;
- **individua errori ortografici:** gli errori vengono sottolineati di rosso;

- **lingua predefinita:** italiano.

Le seguenti convenzioni devono essere rispettate nella stesura dei documenti:

- **grassetto:** deve essere utilizzato nei seguenti casi:
 - titoli di sezioni e paragrafi;
 - termini di elenchi puntati per i quali si fornisce una descrizione;
 - riferimenti alle revisioni di avanzamento.
- **corsivo:** deve essere utilizzato nei seguenti casi:
 - nome del *gruppo*;
 - nome del proponente;
 - nome del progetto;
 - citazioni;
 - abbreviazioni;
 - parole presenti nel glossario;
 - ruoli del progetto;
 - nomi dei documenti.
- **monospace:** deve essere utilizzato nei seguenti casi:
 - nomi di file;
 - codice di programmazione;
 - indirizzi email.
- **maiuscolo:** le uniche parole che possono essere scritte interamente a caratteri maiuscoli sono gli acronimi e le sigle.

3.1.4.3 Titoli

I titoli delle sezioni e dei paragrafi vanno scritti con solo la prima lettera maiuscola a meno di nomi propri e di termini indicati nella sezione [3.1.4.5](#).

3.1.4.4 Elenchi puntati

Le seguenti convenzioni devono essere rispettate nella creazione di elenchi puntati:

- ogni elemento dell'elenco deve iniziare con la lettera minuscola a meno che non sia un nome proprio;
- ogni elemento dell'elenco tranne l'ultimo deve terminare con un punto e virgola;
- l'ultimo elemento dell'elenco deve terminare con il punto.

3.1.4.5 Formati

I seguenti formalismi devono essere utilizzati durante la stesura dei documenti:

- **date:** le date presenti nei documenti devono seguire lo standard *ISO_G* 8601:2004 (vedi riferimento 1.4.1):

YYYY-MM-GG

dove:

- **YYYY:** rappresenta l'anno espresso con quattro cifre;
- **MM:** rappresenta il mese espresso con due cifre;
- **GG:** rappresenta il giorno espresso con due cifre.

È possibile utilizzare il comando $\text{\LaTeX}\ \text{\textbackslash frmdata}\{\text{GG}\}\{\text{MM}\}\{\text{YYYY}\}$ per la formattazione delle date.

- **orari:** gli orari presenti nei documenti devono seguire lo standard *ISO_G* 8601:2004 (vedi riferimento 1.4.1):

hh:mm

dove:

- **hh:** rappresentano le ore espresse con due cifre da 00 a 23;
- **mm:** rappresentano i minuti espressi con due cifre da 00 a 59.

È possibile utilizzare il comando $\text{\LaTeX}\ \text{\textbackslash frmora}\{\text{hh}\}\{\text{mm}\}$ per la formattazione delle ore.

- **valute:** le valute presenti nei documenti devono essere scritte utilizzando il simbolo della valuta usata seguito dal numero. Le cifre decimali devono essere separate dalla virgola, le cifre non decimali devono essere separate da un punto a gruppi di tre:
[Simbolo valuta] 1.234.567,89

Ad esempio: € 3.869,25

- **nomi ricorrenti:** I seguenti termini ricorrenti vanno sempre inseriti con i relativi comandi \LaTeX forniti dal template per garantire omogeneità in tutti i documenti:
 - nome del *gruppo_G*;
 - email di riferimento del *gruppo_G*;
 - nome del proponente;
 - nome del progetto svolto;
 - ruoli di progetto;
 - nomi dei documenti senza versione;
 - nomi dei documenti con versione;
 - revisioni di avanzamento del progetto.

3.1.4.6 Nomi propri

I nomi propri di persona devono essere scritti come nome e cognome.

3.1.5 Componenti grafiche

3.1.5.1 Tabelle

Tutte le tabelle devono avere un indice numerico univoco che le identifichi all'interno del documento ed una breve didascalia. Le tabelle devono essere centrate orizzontalmente.

3.1.5.2 Immagini

Tutte le immagini inserite all'interno di un documento devono avere ampi margini orizzontali che le separino in modo netto dai paragrafi precedenti e successivi per migliorare la leggibilità. Le immagini devono essere centrate orizzontalmente e devono avere larghezza fissa. I diagrammi [UML_G](#) devono essere inseriti nei documenti come immagini.

3.1.6 Classificazione dei documenti

3.1.6.1 Documenti informali

Tutti i documenti non ancora approvati dal responsabile di progetto sono da ritenersi informali e pertanto ad uso unicamente interno.

3.1.6.2 Documenti formali

Un documento diventa formale in seguito all'approvazione da parte del *Responsabile di progetto*. Solo i documenti formali possono essere distribuiti all'esterno del [gruppo_G](#). Prima di poter essere approvato un documento deve essere verificato come descritto nella sezione [3.1.7.2](#) e secondo le procedure descritte nella sezione [3.2](#).

3.1.6.3 Verbali

Per ogni incontro deve essere nominato un segretario che si occuperà della stesura di un verbale. Il verbale deve contenere i seguenti punti:

- **estremi della riunione:**
 - data;
 - ora inizio;
 - ora fine;
 - luogo dove si è svolto l'incontro;
 - lista dei partecipanti;
 - lista degli assenti con eventuali motivazioni;
 - nome del segretario.
- **ordine del giorno:** elenco degli argomenti che saranno discussi;
- **corpo del verbale:** verbale dell'incontro;
- **decisioni prese:** elenco delle decisioni prese identificate in modo univoco.

Una volta approvato dal *Responsabile di progetto* il verbale deve essere distribuito a tutti i componenti del [gruppo_G](#), nel caso di verbale esterno dovrà essere inviato anche al proponente. I nomi dei file dei verbali devono rispettare il seguente formato:

Verbale[Tipologia]_[ID]_[Data riunione].pdf

dove:

- **tipologia:** Esterno o Interno;
- **ID:** identificativo numerico, si distingue tra verbali interni ed esterni, parte da 1;
- **data riunione:** data in cui si è svolta la riunione in formato YYYYMMGG.

I nomi delle decisioni devono rispettare il seguente formato:

V[ID tipologia]_[ID].[Numero decisione]

dove:

- **ID Tipologia:** I o E che significano rispettivamente Interno o Esterno;
- **ID:** identificativo numerico, si distingue tra verbali interni ed esterni, parte da 1;
- **Numero decisione:** numero crescente univoco per quantificare la decisione.

Per la redazione dell'elenco delle decisioni è necessario utilizzare i comandi \LaTeX

- `\itemVE` - per i verbali esterni;
- `\itemVI` - per quelli interni.

3.1.6.4 Versionamento

Tutti i documenti formali ed informali devono essere identificati da una versione, ad ogni nuova versione deve corrispondere una riga nel registro delle modifiche. La versione corrente di un documento deve sempre essere riportata all'interno del documento stesso e va inoltre indicata in coda al nome del file con il seguente formato:

NomeDocumento_vX.Y.Z.pdf

dove:

- **X:**
 - inizia da 0;
 - viene incrementato solo dal *Responsabile di progetto* all'approvazione del documento;
 - non può essere maggiore del numero di revisioni.
- **Y:**
 - inizia da 0;
 - viene incrementato solo dai *Verificatori* ad ogni verifica eseguita;
 - quando viene incrementato X, viene riportato a 0.
- **Z:**
 - inizia da 0;
 - viene incrementato solo dai redattori al completamento di ogni task di modifica del documento;
 - quando vengono incrementati X o Y, viene riportato a 0.

3.1.7 Procedure

3.1.7.1 Creazione di un documento

Viene fornito nel [repository_G](#) un file \LaTeX generico denominato `templateDoc.tex` che contiene il template da utilizzare. Per creare un nuovo documento è sufficiente copiare il file ed inserire il contenuto necessario. Ogni redattore è tenuto a scaricare e mantenere aggiornati tutti i file necessari al template presenti nel [repository_G](#).

3.1.7.2 Ciclo di vita dei documenti

Tutti i documenti tranne i verbali devono seguire il [ciclo di vita_G](#) descritto di seguito (vedi anche figura 1):

1. al termine della stesura di un documento, i redattori ne richiedono la verifica al *Responsabile*;
2. il *Responsabile* assegna la verifica ad un *Verificatore*;
3. il *Verificatore* segnala eventuali modifiche o correzioni da eseguire ai redattori;
4. quando il *Verificatore* ritiene che il documento sia pronto per l'approvazione la richiede al *Responsabile*;
5. il *Responsabile* approva il documento rendendolo formale o lo rifiuta fornendo le motivazioni.

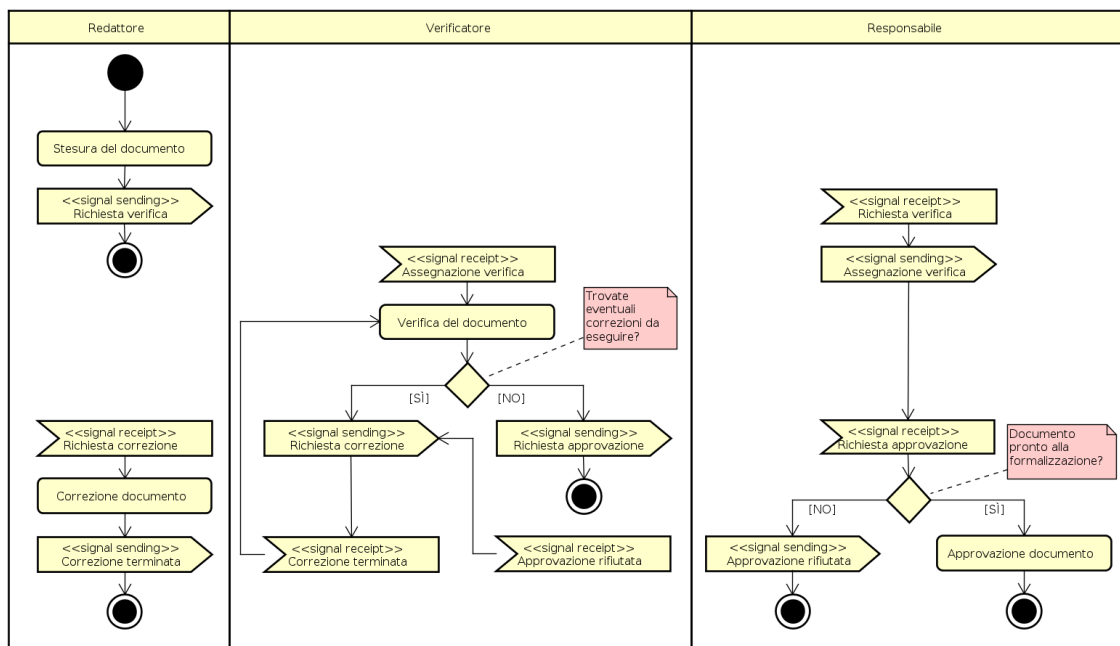


Figura 1: [Ciclo di vita_G](#) di un documento. Riferita nella sezione [3.1.7.2](#)

3.1.8 Strumenti

3.1.8.1 Latex

Per la stesura dei documenti è stato scelto il *linguaggio di markup* \LaTeX . Questo permette di preparare documenti formali divisi in sezioni facilitando la collaborazione tra più editori, di separare il contenuto dalla formattazione grafica e di gestire in maniera automatica vari elementi del testo.

3.1.8.2 TeXstudio

TeXstudio è un editor per la creazione di documenti in \LaTeX . Questo software offre un ambiente di lavoro completo per aiutare la stesura dei documenti. Inoltre integra un compilatore e visualizzatore PDF per il documento prodotto. Tra le principali funzionalità sono presenti:

- evidenziazione della sintassi;
- strumenti per il controllo ortografico;
- completamento automatico.

Questo strumento è utilizzato durante le attività di analisi e progettazione per redigere i documenti necessari. La versione in uso è la 2.11.0 o superiore.

Le principali motivazioni che hanno portato il *gruppo* alla scelta di questo strumento sono:

- gratuito;
- *cross-platform*;
- già conosciuto da alcuni membri del *gruppo*.

Indirizzo per il download:

<http://www.texstudio.org>

3.2 Verifica

3.2.1 Scopo

Lo scopo del processo di verifica è di garantire che ogni attività dei processi svolti non introduca errori nel prodotto e che soddisfi i requisiti o le condizioni necessarie per essere considerata accettabile. La corretta implementazione del processo deve:

- fornire le procedure di verifica necessarie;
- individuare i criteri per la verifica;
- individuare eventuali difetti perchè possano essere corretti.

3.2.2 Procedure

3.2.2.1 Analisi

3.2.2.1.1 Analisi statica

Tecnica utilizzata per l'analisi e la verifica del codice sorgente e della documentazione associata. Può essere applicata secondo due strategie:

- **walkthrough_G**: lettura completa del codice sorgente da analizzare. Va utilizzata unicamente durante le prime **fasi_G** del progetto in quanto risulta onerosa e non efficiente. Gli *Analisti* che la utilizzano devono stilare una lista di controllo con gli errori rilevati più frequentemente;
- **inspection_G**: lettura mirata del codice sorgente da analizzare. È necessario avere una lista di controllo per poter localizzare eventuali punti critici in cui cercare errori. Dopo ogni analisi la lista di controllo deve essere incrementata con eventuali nuovi errori rilevati.

3.2.2.1.2 Analisi dinamica

Tecnica per l'analisi del prodotto software, richiede l'esecuzione del programma e viene eseguita tramite dei test che verificano il funzionamento del prodotto. I test devono essere ripetibili e a parità di condizioni iniziale e ambiente devono fornire lo stesso output. Per ogni test deve quindi essere definito:

- **ambiente**: sistema hardware e software in cui viene eseguito il test;
- **stato iniziale**: stato iniziale da cui si parte ad eseguire il test;
- **input**: input inserito;
- **output**: output atteso.

3.2.2.2 Task di verifica

La gestione dei task di verifica avviene tramite la creazione di ticket nell'applicazione **Teamwork_G** che viene descritta nella sezione 4.1.6.2. Al termine di ogni ticket il *Responsabile di progetto* dovrà seguire la seguente procedura:

1. creare un ticket di verifica con i riferimenti al task da verificare e contrassegnarlo con la dicitura **[VERIFICA]**;
2. assegnare il ticket creato ad un *Verificatore*;
3. al completamento del ticket di verifica contrassegnare il ticket originale con la dicitura **[VERIFICATO]**.

Vedi anche figura 2.

3.2.2.3 Gestione delle anomalie

La gestione delle anomalie durante il processo di verifica avviene tramite la creazione di ticket nell'applicazione **Teamwork_G** che viene descritta nella sezione 4.1.6.2. Nel caso in cui un *Verificatore* dovesse trovare delle anomalie durante un task di verifica queste dovranno essere gestite nel seguente modo:

1. deve essere creato un subticket contrassegnato dalla dicitura **[BUG]** per ogni anomalia riscontrata;
2. assegnare i subticket creati agli assegnatari del task di cui si sta eseguendo la verifica;
3. al completamento di tutti i subticket chiudere il ticket di verifica.

Vedi anche figura 2.

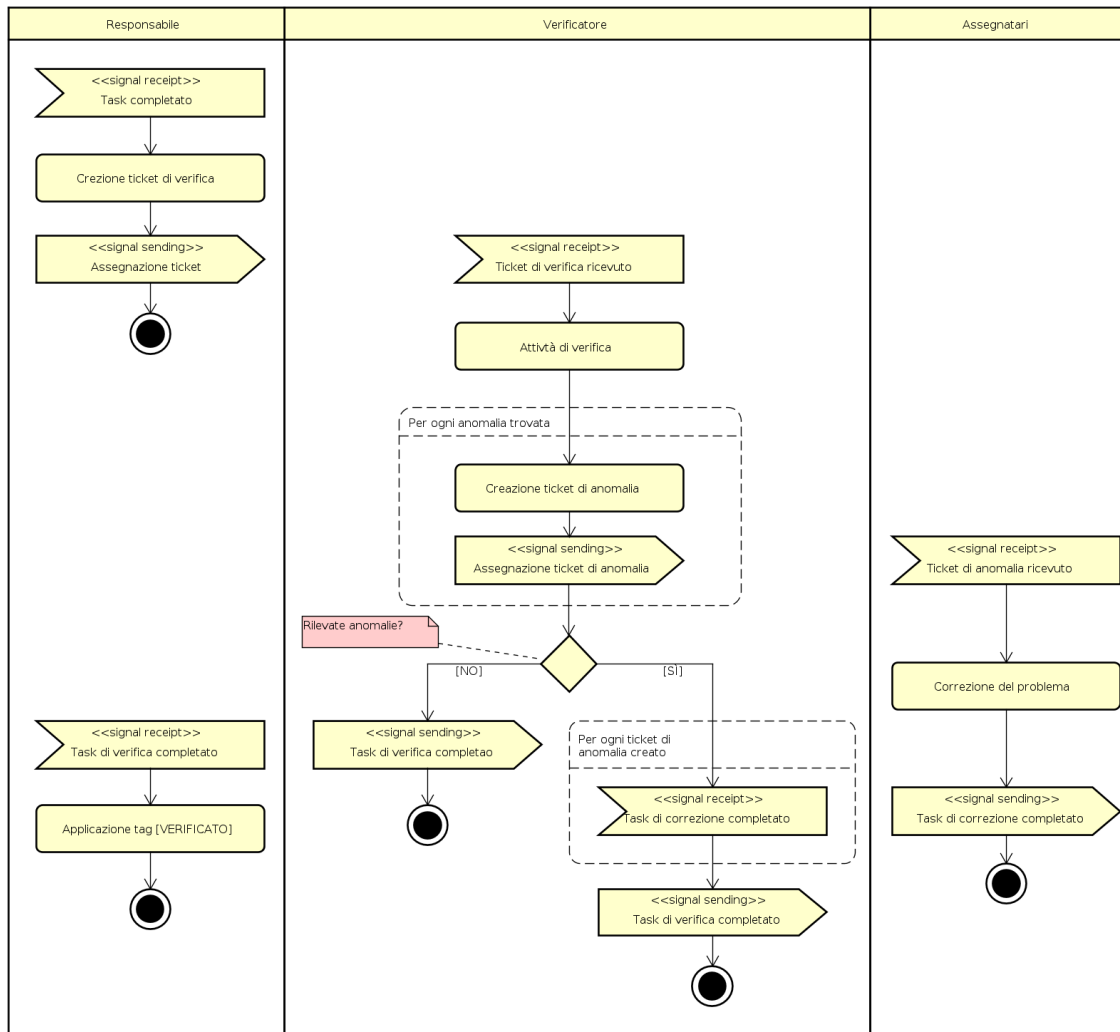


Figura 2: Procedura di verifica dei task e gestione delle anomalie. Riferita nelle sezioni 3.2.2.2 e 3.2.2.3

3.2.3 Strumenti

3.2.3.1 Verifica ortografica

Viene utilizzato il correttore in tempo reale integrato nell'applicazione *TeXstudio_G* descritta nella sezione 3.1.8.2. Il correttore identifica e sottolinea eventuali refusi ortografici; un'analisi più approfondita del testo è compito dei *Verificatori*.

3.2.3.2 Indice leggibilità

La valutazione dell'indice di leggibilità è fatta secondo l'*indice Gulpease_G* utilizzando uno script creato appositamente e fornito assieme al template. Lo script può analizzare sia il file sorgente scritto in $\text{T}_{\text{E}}\text{X}$ che il file pdf risultante.

3.2.3.3 Validazione web

Eventuali test di validazione di pagine web devono essere eseguiti utilizzando

<https://validator.w3.org/>

3.3 Validazione

3.3.1 Scopo

Lo scopo del processo di validazione è di determinare in maniera oggettiva che il prodotto esaminato sia conforme ai requisiti richiesti e che soddisfi il compito per cui è stato creato. La corretta implementazione del processo deve:

- utilizzare gli stessi strumenti e le stesse procedure del processo di verifica;
- fornire tutti i dati necessari alla valutazione del prodotto;
- verificare che tutte le metriche stabilite siano soddisfatte.

4 Processi organizzativi

4.1 Gestione processi

4.1.1 Scopo

Lo scopo della gestione dei processi è di migliorare l'organizzazione e la cooperazione tra i membri del *gruppo_G*. La corretta implementazione del processo deve:

- stabilire le modalità di comunicazione del *gruppo_G*;
- definire i ruoli ed i compiti specifici;
- fornire la documentazione su strumenti di organizzazione e relative procedure.

4.1.2 Comunicazioni

4.1.2.1 Interne

Per la comunicazione interna dei membri del *gruppo_G*, viene utilizzata l'applicazione *Slack_G* descritta in maniera più dettagliata nella sezione 4.1.6.1.

4.1.2.2 Esterne

Per le comunicazioni esterne è stata creata la seguente mail:

zephyrus.swe@gmail.com

Il *Responsabile di progetto* è la persona incaricata a inviare comunicazioni con questo indirizzo. Tutte le mail ricevute verranno inoltrate automaticamente ad ogni membro del *gruppo_G* a titolo informativo.

4.1.3 Incontri

4.1.3.1 Interni

Il *Responsabile di progetto* ha il compito di organizzare gli incontri interni rispettando la procedura descritta nella sezione 4.1.5.1. Qualsiasi membro del *gruppo_G* può richiedere un incontro interno. Sarà compito del *Responsabile di progetto* accettare o rifiutare la richiesta. Al termine dell'incontro deve essere redatto un verbale, di cui si rimanda alla sezione 3.1.6.3 per la sua composizione in dettaglio.

4.1.3.2 Esterni

Il *Responsabile di progetto* ha il compito di organizzare gli incontri esterni con il proponente o committente seguendo la procedura descritta nella sezione 4.1.5.2. Qualsiasi membro del *gruppo_G* può richiedere un incontro esterno. Sarà compito del *Responsabile di progetto* accettare o rifiutare la richiesta. Al termine dell'incontro deve essere redatto un verbale, di cui si rimanda alla sezione 3.1.6.3 per la sua composizione in dettaglio.

4.1.4 Ruoli di progetto

Ogni componente del *gruppo_G* deve ricoprire almeno una volta ciascuno dei ruoli previsti nello sviluppo del progetto. Nel *Piano di progetto* vengono assegnati compiti e ruoli che i membri del *gruppo_G* si impegnano a rispettare. Di seguito sono elencati i diversi incarichi, delineando per ciascuno mansioni e responsabilità.

4.1.4.1 Responsabile di progetto

Il *Responsabile di progetto* è la figura che rappresenta il *gruppo_G* e il progetto presso committente e proponente. Approva le scelte prese dal *gruppo_G* e se ne assume la responsabilità. La sua presenza segue tutta la durata del progetto. Le sue responsabilità sono:

- gestione delle risorse;
- approvazione della documentazione di progetto;
- analisi e mitigazione dei rischi;
- coordinamento e pianificazione delle attività di progetto seguendo il *Piano di progetto*.

4.1.4.2 Amministratore di progetto

L'*Amministratore* è responsabile dell'ambiente di lavoro del *gruppo_G*, ne controlla l'efficienza e l'operatività. Le sue principali responsabilità sono:

- controllo delle versioni e configurazioni del prodotto;
- gestione del versionamento e dell'archiviazione della documentazione;
- risoluzione dei problemi inerenti la gestione di processi e risorse;
- controllo e miglioramento degli strumenti di lavoro e dell'infrastruttura;
- redazione e aggiornamento delle *Norme di progetto*.

4.1.4.3 Analista

L'*Analista* ha il compito di esaminare e studiare attentamente il dominio del problema, la sua presenza non è necessaria per tutta la durata del progetto. Le sue responsabilità sono:

- capire il dominio di lavoro del cliente;
- analizzare e capire la natura del problema posto dal cliente;
- redigere lo studio di fattibilità e l'analisi dei requisiti.

4.1.4.4 Progettista

Il *Progettista* è responsabile di tutto ciò che riguarda la progettazione software. Deve avere conoscenze tecniche e tecnologiche aggiornate per la gestione del progetto. Le sue responsabilità sono:

- fornire una soluzione attuabile entro i limiti di tempo;
- descrivere il funzionamento del sistema a diversi livelli di dettaglio;
- effettuare scelte su aspetti tecnici del progetto, rendendolo efficiente, robusto e manutenibile.

4.1.4.5 Programmatore

Il *Programmatore* si occupa dell'attività di codifica. Le sue responsabilità sono:

- implementare le scelte dettate dal *Progettista*, senza apportare modifiche personali;
- documentare il codice prodotto;
- rispettare le convenzioni riportate nel presente documento;
- realizzare strumenti per verifica e validazione.

4.1.4.6 Verificatore

Il *Verificatore* è responsabile dell'attività di verifica. Deve avere una profonda conoscenza delle *Norme di progetto* ed è presente per tutta la durata del progetto. Le sue responsabilità sono:

- controllare l'osservazione delle *Norme di progetto* lungo tutte le attività del progetto stesso.

4.1.5 Procedure

4.1.5.1 Organizzazione incontri interni

L'organizzazione degli incontri interni deve essere svolta rispettando il seguente ordine:

1. controllare il foglio condiviso su [Google Drive_G](#) *Orari impegni ricorrenti*;
2. postare sul canale [Slack_G](#) *#incontri-interni* alcune proposte di incontro, indicando: luogo, data e orario;
3. assicurarsi che tutti i membri del [gruppo_G](#) abbiano espresso la preferenza, in caso contrario entro 24 ore, contattarli telefonicamente;
4. aggiungere l'evento nel calendario di [Teamwork_G](#) con le seguenti caratteristiche:
 - **titolo evento:** Riunione interna;
 - **dove:** data, ora inizio, ora fine;
 - **dettagli:** luogo concordato, breve descrizione degli argomenti da trattare;
 - **persone:** assegnare i partecipanti;
 - **reminders:** assegnare dei promemoria se lo si ritiene necessario.

4.1.5.2 Organizzazione incontri esterni

L'organizzazione degli incontri esterni deve essere svolta rispettando il seguente ordine:

1. contattare il proponente per avere informazioni sulla sua disponibilità;
2. postare sul canale [Slack_G](#) *#incontri-esterni* alcune proposte di incontro, indicando: luogo, data e orario;
3. assicurarsi che tutti i membri del [gruppo_G](#) abbiano espresso la preferenza, in caso contrario entro 24 ore, contattarli telefonicamente;
4. aggiungere l'evento nel calendario di [Teamwork_G](#) con le seguenti caratteristiche:

- **Titolo evento:** Riunione esterna;
- **Dove:** data, ora inizio, ora fine;
- **Dettagli:** luogo concordato, breve descrizione degli argomenti da trattare;
- **Persone:** assegnare i partecipanti;
- **Reminders:** assegnare dei promemoria se lo si ritiene necessario.

4.1.5.3 Gestione dei ticket

Accedere allo spazio *TeamworkG* del *gruppoG*, posizionandosi nella sezione "All Task" oppure tramite il seguente link: <https://swe2016.teamwork.com/#projects/140646/tasks>. Premere su "Add Task" per la creazione o sul nome di un task già esistente per la modifica.

4.1.5.3.1 Creazione di un ticket

La creazione di un ticket deve essere svolta rispettando il seguente ordine:

1. inserimento titolo task;
2. assegnazione delle persone incaricate al suo svolgimento;
3. inserimento data prevista di completamento;
4. inserimento descrizione;
5. inserimento priorità;
6. selezionare eventuali dipendenze da altri task;
7. inserimento di un *tagG* che identifichi il ruolo delle persone incaricate;
8. salvataggio task.

Vedi figura 3.

4.1.5.3.2 Modifica di un ticket

La modifica di un ticket deve essere svolta rispettando il seguente ordine:

1. ricerca del ticket da modificare;
2. modifica di uno o più campi;
3. verifica inserimento dei campi dati: titolo, assegnatario, descrizione e data di completamento;
4. salvare le modifiche apportate.

Vedi figura 4.

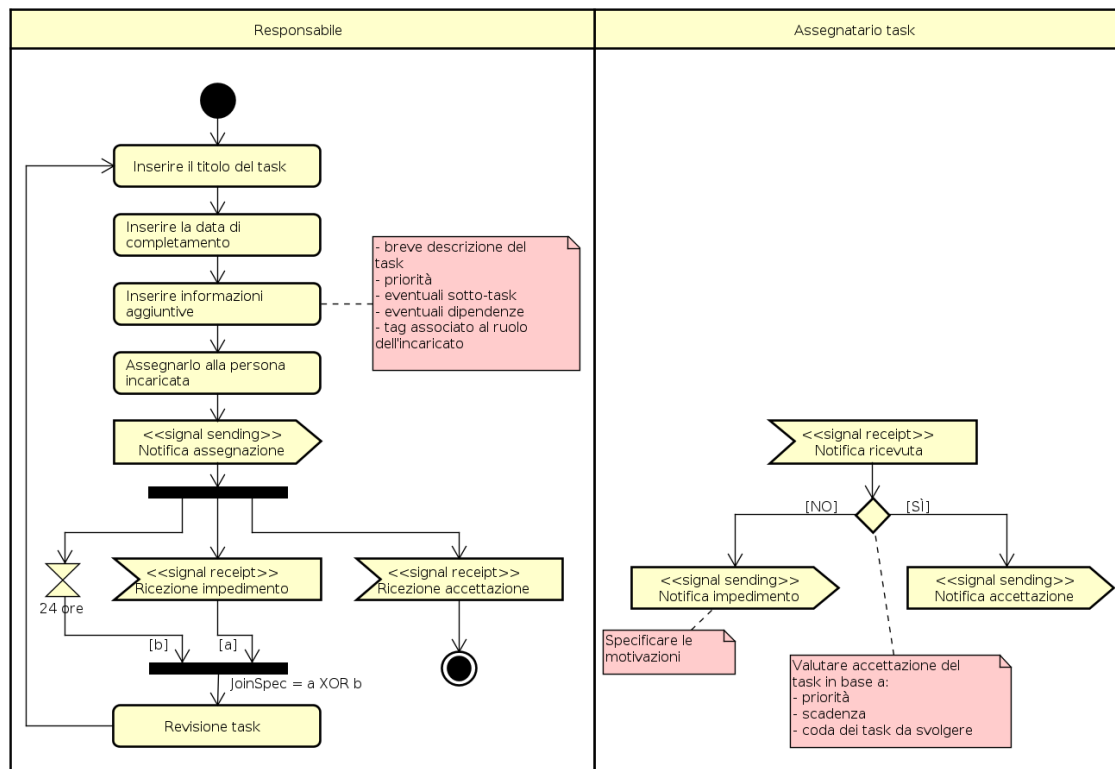


Figura 3: Procedura assegnazione ticket. Riferita nella sezione 4.1.5.3.1

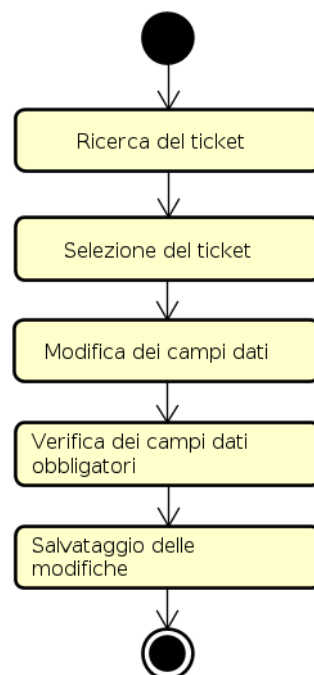


Figura 4: Procedura modifica ticket. Riferita nella sezione 4.1.5.3.2

4.1.5.4 Gestione delle milestone

Accedere allo spazio *Teamwork_G* del *gruppo_G*, posizionandosi nella sezione "Milestones" oppure tramite il seguente link: <https://swe2016.teamwork.com/#/projects/140646/milestones/upcoming>. Premere su "Add *Milestone_G*" per la creazione o sul nome di una *milestone_G* esistente per la modifica.

4.1.5.4.1 Creazione milestone

La creazione di una *milestone_G* deve essere svolta rispettando il seguente ordine:

1. inserimento titolo *milestone_G*;
2. inserimento data;
3. assegnazione persone coinvolte;
4. assegnazione reminders;
5. inserimento descrizione;
6. salvataggio *milestone_G*.

Vedi figura 5.

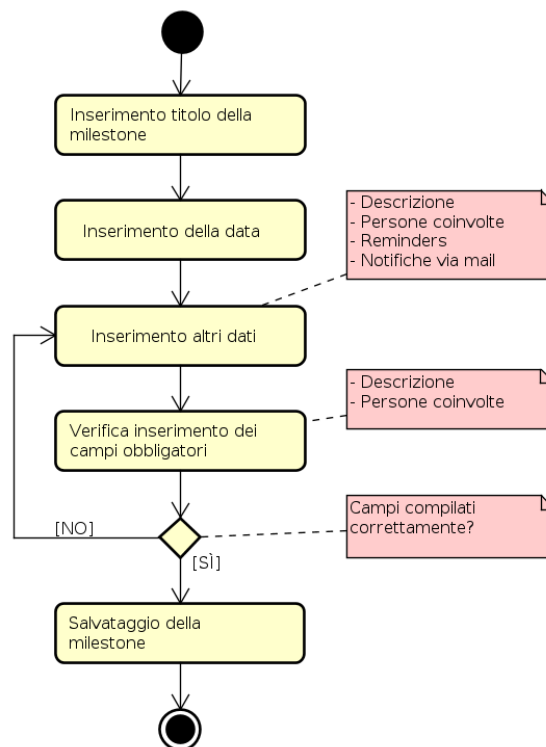


Figura 5: Procedura inserimento *milestone_G*. Riferita nella sezione 4.1.5.4.1

4.1.5.4.2 Modifica milestone

La modifica di una *milestone_G* deve essere svolta rispettando il seguente ordine:

- selezione *milestone_G*;
- modifica di uno o più campi dati;
- verifica inserimento dei campi dati: titolo, data, descrizione e assegnatari;
- salvataggio *milestone_G*.

Vedi figura 6.

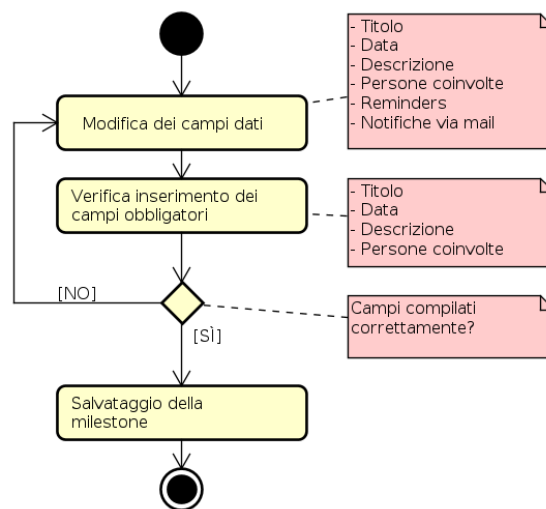


Figura 6: Procedura modifica *milestone_G*. Riferita nella sezione 4.1.5.4.2

4.1.6 Strumenti

4.1.6.1 Slack

Slack_G è un servizio gratuito di messaggistica professionale disponibile su piattaforme mobile, desktop e web. Le principali motivazioni che hanno portato il *gruppo_G* alla scelta di questo strumento sono:

- possibilità di integrazione con molti servizi (tra i principali: *Dropbox_G*, *GitHub_G*, *GoogleDrive*, bot etc.);
- possibilità di creazione di canali tematici personalizzati con la profilazione delle utenze e delle notifiche;
- utilizzato come ulteriore canale per comunicare con *RiskApp* in quanto anche loro lo utilizzano;
- possibilità di richiamare all'attenzione un membro del *gruppo_G* con il comando "@";
- possibilità di classificare un commento come "rilevante" tramite il comando "pin to";
- possibilità di inserire dei reminder sui messaggi;

- *cross-platform_G* e *cross-device_G*;
- interfaccia più ricca e organizzata rispetto alle usuali applicazioni di messaggistica.

Lo spazio dedicato al *gruppo_G* si trova al seguente indirizzo:

<https://zephyrus-swe.slack.com/home>

4.1.6.2 Teamwork

Teamwork_G è un'applicazione web di project management che permette di sfruttare le seguenti funzionalità principali:

- gestione dei task;
- gestione degli appuntamenti con scadenze a calendario;
- pianificazione del lavoro;
- rendiconto ore lavoro su intero progetto e/o su specifici task.

Le principali motivazioni che hanno portato il *gruppo_G* alla scelta di questo strumento sono:

- le funzionalità essenziali sono gratuite;
- alta portabilità ed accessibilità essendo fruibile via web;
- interfaccia semplice e funzionale.

Lo spazio di lavoro dedicato al *gruppo_G* si trova al seguente indirizzo:

<https://swe2016.teamwork.com/>

4.1.6.3 Condivisione file

Dropbox_G è un servizio che offre la possibilità di salvare file su una piattaforma *cloud_G* personale e mantenerli sincronizzati tra diversi dispositivi tramite un client. Il *gruppo_G* ha scelto *Dropbox_G* per gestire file che non necessitano versionamento, inoltre le principali motivazioni che ne hanno portato alla scelta sono:

- alta velocità nella sincronizzazione dei file;
- servizio già conosciuto e largamente utilizzato da tutti i membri del *gruppo_G*;
- lo spazio disponibile nella versione gratuita è sufficiente per consentire una discreta quantità di file associati a questo progetto;
- alta portabilità e usabilità essendo *cross-platform_G* e *cross-device_G*.

Indirizzo per il download:

<https://www.dropbox.com>

Google Drive_G è un servizio *cloud_G*, di memorizzazione e sincronizzazione online offerto da Google. Il servizio comprende il *file hosting_G*, il file sharing e la modifica collaborativa di documenti. Per accedere allo spazio condiviso è necessario che ogni membro sia prima aggiunto dal *Responsabile di progetto*. Il *gruppo_G* ha scelto questo strumento per la possibilità di redigere documenti in maniera collaborativa come attività preliminare a fine organizzativo. Le principali motivazioni che hanno portato il *gruppo_G* alla scelta di questo strumento sono:

- servizio già conosciuto e utilizzato da tutti i membri del *gruppo_G*;
- gratuito;
- alta portabilità e usabilità essendo *cross-platform_G* e *cross-device_G*.

Lo spazio di lavoro dedicato al *gruppo_G* di trova al seguente link:

[Google Drive Zephyrus](#)

4.2 Gestione delle infrastrutture

4.2.1 Scopo

Lo scopo del processo di gestione delle infrastrutture è di stabilire e mantenere le infrastrutture e gli strumenti necessari allo svolgimento dei processi durante lo svolgimento del progetto. La corretta implementazione del processo deve:

- fornire un ambiente di lavoro idoneo;
- garantire il funzionamento di tutte le infrastrutture necessarie;
- indicare il corretto utilizzo delle infrastrutture.

4.2.2 Struttura dei repository

Il *gruppo_G* ha scelto di utilizzare *GitHub_G* per il versionamento e il salvataggio dei file inerenti le attività di progetto. L'Amministratore di progetto deve creare e organizzare i *repository_G* necessari e assicurarsi che tutti i membri del *gruppo_G* vi possano accedere. Ogni membro deve essere registrato su *GitHub_G* e aver attivato l'account studente.

Il *repository_G* dedicato ai documenti si trova al seguente indirizzo:

<https://github.com/JordanGottardo/Documenti>

Il *repository_G* dedicato al codice si trova al seguente indirizzo:

<https://github.com/JordanGottardo/DeGeOp>

Le cartelle nel *repository_G* vengono organizzate nel seguente modo a partire dalla root:

- **documenti:** sono presenti le cartelle per ogni *fase_G* del progetto:
 - **01-RR:** contenente i documenti e i file necessari alla **Revisione dei requisiti**;
 - **02-RP:** contenente i documenti e i file necessari alla **Revisione di progettazione**;
 - **03-RQ:** contenente i documenti e i file necessari alla **Revisione di qualifica**;
 - **04-RA:** contenente i documenti e i file necessari alla **Revisione di accettazione**;
 - **Script:** contenente script e altri strumenti utili per la stesura e la verifica dei documenti;
 - **Template:** contenente i file del template da usare per la creazione dei documenti \LaTeX .
- **codice:** la descrizione di questa cartella verrà fornita in una *fase_G* successiva di lavoro.

4.2.3 Nomi dei file

I nomi dei documenti presenti nel *repository_G* devono rispettare la notazione *camel case_G* con le seguenti caratteristiche:

- la prima lettera di ogni file è maiuscola, le successive minuscole fino al presentarsi della parola successiva che inizia con la lettera maiuscola e così via;
- è possibile utilizzare unicamente caratteri alfanumerici e il trattino basso;
- i nomi non possono contenere spazi o elementi di punteggiatura.

4.2.4 Messaggi di commit

Ogni volta che si effettuano modifiche sui file del *repository_G* locale per poi esser caricate in quello remoto, bisogna specificarne le motivazioni. Per uniformare l'ambiente di lavoro è stato scelto un formato standard per la scrittura di una *commit_G*. Si veda la sezione 4.2.5.2 per il dettaglio sulla procedura da seguire.

4.2.5 Procedure

4.2.5.1 Installazione e configurazione di Git

L'installazione di *Git_G* varia a seconda del sistema operativo utilizzato. Di seguito verranno elencate le principali procedure d'installazione.

Per i sistemi *Linux_G*:

- aprire il terminale;
- eseguire il comando `sudo apt-get update`;
- eseguire il comando `apt-get install gitG`.

Per i sistemi *Windows_G*:

- accedere al sito ufficiale <https://git-scm.com/download/win>;
- scaricare l'eseguibile;
- installare l'eseguibile seguendo la procedura guidata.

Per i sistemi *MacOS_G*:

- accedere al sito ufficiale <https://git-scm.com/download/mac>;
- scaricare l'eseguibile;
- aprire il file appena scaricato e avviare l'installazione cliccando sul file `.pkg`.

La configurazione iniziale prevede l'inserimento del nome dell'utente e la connessione con l'account *GitHub_G* tramite la seguente procedura:

- `git config --global user.name <Nome Cognome>`: imposta il nome dell'utente, che comparirà come autore delle *commit_G* effettuate;
- `git config --global user.email <email>`: imposta l'email dell'utente; deve essere impostata la stessa email utilizzata per la registrazione su *GitHub_G*.

Per la creazione di una cartella locale del *repository_G* si deve seguire la seguente procedura:

- creare una nuova cartella;
- aprire il terminale;
- posizionarsi all'interno della cartella precedentemente creata;
- eseguire il comando: `git init`;
- recuperare l'indirizzo URL del progetto su *GitHub_G*;
- eseguire il comando: `git clone <indirizzo appena recuperato>`.

4.2.5.2 Invio di commit

Il comando *commit_G* adibito all'esecuzione dell'omonima operazione, deve essere composto da un messaggio riassuntivo delle operazioni svolte e una breve descrizione contenente:

- lista dei file coinvolti;
- lista delle modifiche effettuate per ogni file.

4.2.5.3 Comandi utili Git

Per l'approfondimento dei principali comandi riguardanti l'utilizzo di *Git_G* è stata redatta una guida, consultabile al seguente indirizzo: [Guida Git](#).

4.2.6 Strumenti

4.2.6.1 Git

Git_G è un *sistema software di controllo di versione_G* distribuito e *open source_G*. La versione utilizzata al momento della stesura di questo documento è la 2.11.0 o superiore. Le principali motivazioni che hanno portato il *gruppo_G* alla scelta di questo strumento sono:

- largamente usato in ambito lavorativo;
- performance superiori rispetto ad altri sistemi di versionamento;
- sistema distribuito anziché centralizzato.

Indirizzo per il download e documentazione:

<https://git-scm.com>

L'utilizzo di *Git_G* verrà effettuato tramite riga di comando. Si lascia libertà ai membri del *gruppo_G* per l'installazione di eventuali interfacce grafiche personalizzate.

4.2.6.2 GitHub

[GitHub_G](#) è un servizio di web di hosting per lo sviluppo di progetti software. Tra le caratteristiche principali:

- utilizzo del [sistema di controllo di versione_G Git_G](#);
- oltre al codice sorgente è possibile inserire documentazione e immagini;
- [issue_G](#) tracking;
- funzionalità simili ai social network come follower, commenti e notifiche;
- visione di grafici e statistiche su sviluppatori e [repository_G](#).

Le principali motivazioni che hanno portato il [gruppo_G](#) alla scelta di questo strumento sono:

- possibilità di creare [repository_G](#) private attivando il piano studente con l'email universitaria;
- sistema largamente diffuso e conosciuto da alcuni membri del [gruppo_G](#);
- integrabile con altre applicazioni (es: [Slack_G](#), [IDE_G](#)).

4.2.6.3 Sistemi operativi

I membri del [gruppo_G](#) operano sui seguenti sistemi operativi:

- [Linux_G](#) distribuzioni: Debian v8.0, Ubuntu v16.04 e superiori, Mint v17.1;
- [Windows_G](#) versione: 10;
- [MacOS_G](#) versione: 10.12.

4.3 Apprendimento

4.3.1 Scopo

Lo scopo del processo di apprendimento è di garantire che tutti i membri del [gruppo_G](#) abbiano conoscenze e capacità sufficienti per svolgere le attività assegnatagli. Nel caso in cui un componente del [gruppo_G](#) ritenga di non essere in grado di svolgere un task dovrà segnalarlo immediatamente al *Responsabile di progetto* che dovrà organizzare le attività necessarie all'apprendimento.

A Lista di controllo

Durante l'applicazione del *walkthrough_G* ai documenti, sono riportati di seguito gli errori più frequenti. Per migliorare l'efficienza e l'efficacia da parte dei verificatori è opportuno che si basino sui seguenti controlli:

- **lingua italiana:**
 - la prima parola di una voce dell'elenco puntato inizia con la lettera maiuscola;
 - la voce finale dell'elenco puntato non termina con il punto;
 - una voce intermedia dell'elenco puntato non termina con il punto e virgola.
- **norme stilistiche:**
 - il carattere "e maiuscolo accentato" è scritto E' invece di È;
 - i due punti in grassetto dopo un termine in grassetto.
- **LaTeX:**
 - date e orari non scritti con i rispettivi comandi `\frmdata{GG}{MM}{YYYY}` e `\frmora{hh}{mm}`;
 - mancato utilizzo dei comandi personalizzati;
 - utilizzo scorretto delle parentesi graffe dopo i comandi LaTeX;
 - mancato aggiornamento dell'intestazione del documento dopo una modifica;
 - link e riferimenti non funzionanti o assenti.
- **UML_G:**
 - casi d'uso non proporzionati correttamente tra loro;
 - collegamenti in uscita non ad angolo retto.
- **glossario:**
 - mancata evidenziazione di termini presenti nel *Glossario*;
 - termini evidenziati impropriamente non presenti nel *Glossario*.
- **nomi dei documenti:**
 - mancata indicazione della versione di riferimento di un documento.