

21 dicembre 2015

StarWare

Quizzipedia: software per la gestione di questionari



Norme di Progetto

Informazioni sul documento

Nome Documento	Norme di Progetto
Versione	1.0.3
Stato	<i>Informale</i>
Uso	<i>Interno</i>
Data Creazione	30 novembre 2015
Data Ultima Modifica	21 dicembre 2015
Redazione	Nicola De Cao Andrea Venier Alessio Vitella
Verifica	Igor Baylyak Thomas Pigarelli
Approvazione	Anna Bonaldo
Lista Distribuzione	StarWare

Registro delle modifiche

Versione	Autore	Data	Descrizione
1.0.6	Alessio Vitella	2015-12-21	Versioning e Repository
1.0.5	Andrea Venier	2015-12-10	Ticketing
1.0.4	Andrea Venier	2015-12-08	Ruoli di progetto e Riunioni
1.0.3	Alessio Vitella	2015-12-07	Norme del Piano di Progetto e dell'Analisi dei requisiti
1.0.2	Alessio Vitella	2015-12-03	Descrizione processi di base
1.0.1	Nicola De Cao	2015-12-02	Introduzione
1.0.0	Nicola De Cao	2015-11-30	Template di base

Tabella 1: Versionamento del documento

Indice

1	Introduzione	6
1.1	Scopo	6
1.2	Descrizione	6
1.3	Glossario	6
2	Processi base	7
2.1	Processi del fornitore	7
2.1.1	Scopo	7
2.1.2	Descrizione	7
2.1.3	Piano di Qualifica	7
2.1.4	Piano di Progetto	7
2.2	Processi di sviluppo	8
2.2.1	Scopo	8
2.2.2	Descrizione	8
2.2.3	Analisi dei requisiti	8
2.2.3.1	Scopo	8
2.2.3.2	Descrizione	8
2.2.3.3	Casi d'uso	8
2.2.3.4	Classificazione dei Requisiti	9
2.2.4	Progettazione	9
2.2.4.1	Scopo	9
2.2.4.2	Descrizione	9
2.2.4.3	UML	9
2.2.5	Codifica	10
3	Processi di Supporto	11
3.1	Scopo	11
3.2	Descrizione	11
3.3	Documentazione	11
3.3.1	Versione	11
3.3.2	Stato	12
3.3.2.1	Informale	12
3.3.2.2	Formale	12
3.3.3	Uso	12
3.3.3.1	Interno	12
3.3.3.2	Esterno	13
3.3.4	Struttura dei documenti	13
3.3.4.1	Prima pagina	13
3.3.4.2	Diario delle modifiche	13
3.3.4.3	Indice	14
3.3.4.4	Elenco tabelle e figure	14
3.3.4.5	Introduzione	14
3.3.4.6	Sezioni	14
3.3.4.7	Formattazione delle pagine	14
3.3.5	Norme tipografiche	15
3.3.5.1	Stile del testo	15

3.3.5.2	Composizione	15
3.3.5.3	Formati	16
3.3.5.4	Tabelle	17
3.3.5.5	Immagini e grafici	17
3.4	Linguaggio	17
3.5	Configurazione	17
3.5.1	Documentazione	17
3.6	Verifica	18
3.6.1	Tecniche di analisi	18
3.6.2	Classificazione dei problemi	19
3.6.3	Diagrammi UML	20
3.6.4	Documentazione	20
3.7	Validazione	21
3.8	Risoluzione dei Problemi	21
4	Processi di Organizzazione	22
4.1	Processi di Management	22
4.1.1	Scopo	22
4.1.2	Descrizione	22
4.1.3	Ruoli	22
4.1.3.1	Project Manager	22
4.1.3.2	Amministratore	23
4.1.3.3	Analista	23
4.1.3.4	Progettista	24
4.1.3.5	Programmatore	24
4.1.3.6	Verificatore	24
4.1.4	Comunicazione	25
4.1.4.1	Esterna	25
4.1.4.2	Interna	25
4.1.5	Riunioni	25
4.1.5.1	Frequenza	25
4.1.5.2	Convocazione riunione	25
4.1.5.3	Verbale	26
4.1.6	Pianificazione	26
4.1.6.1	Pianificazione delle attività	26
4.1.6.2	Assegnazione dei task	27
4.1.7	Esecuzione e controllo	27
4.2	Processi di gestione dell'infrastruttura	28
4.2.1	Scopo	28
4.2.2	Descrizione	28
4.2.3	Ticketing	28
4.2.3.1	Struttura Ticket	28
4.2.3.2	Etichette	28
4.2.3.3	Ciclo di vita del Ticket	29
4.2.3.4	Ticket di Verifica	29
4.2.4	Versioning	30
4.2.5	Repository	31
4.2.5.1	Comandi	31

4.2.5.2	Struttura albero delle directory	32
---------	--	----

Elenco delle tabelle

1	Versionamento del documento	1
2	Classificazione dei problemi	20

Elenco delle figure

1	Diagramma di gestione dei Ticket.	30
2	Diagramma di gestione dei Ticket di Verifica.	31

1 Introduzione

Questo documento definisce le norme che i membri del gruppo StarWare adotteranno nello svolgimento del progetto Quizzipedia: software per la gestione di questionari. Tutti i membri sono tenuti a leggere il documento attentamente e a seguirne le norme.

1.1 Scopo

Il documento si propone come proposito di garantire l'uniformità di tutto il materiale prodotto, migliorarne l'efficienza e ridurre gli errori.

1.2 Descrizione

Verranno definite norme riguardanti:

- interazioni tra membri del gruppo;
- stesura documenti, convenzioni stilistiche e tipografiche;
- modalità di lavoro durante tutte le fasi del progetto;
- ambiente di lavoro.

1.3 Glossario

Al fine di evitare ogni ambiguità di linguaggio e massimizzare la comprensione dei documenti, i termini tecnici, di dominio, gli acronimi e le parole che necessitano chiarimenti sono riportate nel documento *Glossario_v1.2.0*. Ogni occorrenza di vocaboli presenti nel *Glossario*_G è marcata da una G maiuscola in pedice (e.g. pedice_G).

2 Processi base

I processi base sono formati dai seguenti sotto processi:

1. Processi dell'acquirente
2. Processi del fornitore
3. Processi di sviluppo
4. Processi operativi
5. Processi di manutenzione

Per lo scopo del corrente progetto saranno dettate solo le norme per le fasi del fornitore e di sviluppo, poiché i processi dell'acquirente non riguardano il team e le fasi operative e di manutenzione non verranno trattate.

2.1 Processi del fornitore

2.1.1 Scopo

Definisce le attività e i task attraverso i quali il fornitore comunica al committente modo, tempi e costi previsti per il progetto.

2.1.2 Descrizione

Consiste nel redigere il *Piano di Qualifica_G*, e il *Piano di Progetto_G*.

2.1.3 Piano di Qualifica

Il *Piano di Qualifica_G* dovrà illustrare la strategia complessiva di verifica e validazione e gli strumenti utilizzati per attuarla. Lo scopo è quello di pervenire al collaudo del sistema con la massima efficienza ed efficacia.

2.1.4 Piano di Progetto

Il *Piano di Progetto_G* dovrà presentare l'organigramma dettagliato del fornitore, lo schema proposto per l'assegnazione e la rotazione dei ruoli di progetto, l'impegno complessivo previsto per ogni ruolo e per ogni individuo, e il conto economico preventivo di realizzazione del prodotto.

Il *Project Manager_G*, assieme all'*Amministratore_G*, deve presentare nel *Piano di Progetto_G* diagrammi di Gant e ripartizione delle risorse relativi ad ogni fase del Progetto. In Particolare è necessario che ogni componente del gruppo ricopra tutti i ruoli nell'arco dello svolgimento del progetto. La rotazione dei ruoli deve garantire un'equa ripartizione del carico di lavoro individuale, ovvero il totale di ore produttive per ogni persona può differire al più di poche unità da quello degli altri. Inoltre è possibile che un componente rivesta più ruoli contemporaneamente, ma sempre evitando conflitti di interesse. In particolare i conflitti di interesse da evitare sono quelli tra *Project Manager_G* e qualsiasi altro ruolo o tra *Verificatore_G* e qualsiasi altro ruolo.

2.2 Processi di sviluppo

2.2.1 Scopo

Il processo di sviluppo produce un prodotto software che soddisfi i requisiti architetturali. Inoltre definisce un insieme di azioni che formalizzano comportamenti, interfacce e vincoli di implementazione atti a creare un prodotto software

2.2.2 Descrizione

Il processo di sviluppo è un insieme di attività e task necessari per lo sviluppo di un prodotto software. Coloro che eseguono queste attività' sono gli sviluppatori. Le attività che compongono i processi di sviluppo sono:

1. Processo di Analisi dei requisiti
2. Processo di Progettazione
3. Processo di Codifica

2.2.3 Analisi dei requisiti

2.2.3.1 Scopo

Gli Analisti dovranno produrre l'Analisi dei Requisiti basandosi sul capitolato e sugli incontri con il proponente. Tale processo ha l'obiettivo di formalizzare e rendere tracciabile in un documento i requisiti e casi d'uso individuati, comprendendo a fondo eventuali problemi da risolvere in fase di progettazione.

2.2.3.2 Descrizione

Con il completamento di questo processo si ottiene una documentazione affidabile e consistente che ben descrive le esigenze e le richieste del proponente.

2.2.3.3 Casi d'uso

I casi d'uso sono creati secondo lo standard UML 2.4 e sono identificati dalla seguente notazione:

UC[codice] - [Titolo]

dove il **codice** è un numero progressivo identificativo di ogni requisito, gerarchico nel caso di sotto-casi d'uso tramite la notazione *CodiceUCPadre.CodiceSottoUC*. Per ogni caso d'uso deve inoltre essere indicato:

- **Titolo** breve ma non ambiguo
- **Attori** principali e secondari coinvolti
- **Precondizione**
- **Descrizione** sintetica del Caso d'uso
- **Postcondizione**
- **Requisiti** collegati al Caso d'uso

2.2.3.4 Classificazione dei Requisiti

È compito degli Analisti stilare una lista dei requisiti emersi dal capitolato e da eventuali riunioni con il proponente. Questi requisiti dovranno essere classificati per tipo e importanza, utilizzando la seguente codifica:

R[importanza][tipo][codice]

- **Importanza** può assumere i seguenti valori:
 - 0: Requisito obbligatorio
 - 1: Requisito desiderabile
 - 2: Requisito opzionale
- **Tipo** può assumere i seguenti valori:
 - F: Funzionale, descrive i servizi o le funzioni offerte dal sistema
 - N: Non funzionale, descrivono i vincoli sui servizi offerti dal sistema
- **Codice** è un numero progressivo univoco per ogni requisito, indipendente da importanza e tipo. Nel caso si abbia un sotto-requisito codice può anche essere espresso in modo gerarchico tramite la notazione *CodiceRequisitoPadre.CodiceSottorequisito*

Ogni requisito deve essere correlato da una sintetica ma precisa descrizione. Per ogni requisito bisogna indicarne le fonti, che posso essere il capitolato o uno o più casi d'uso.

2.2.4 Progettazione

2.2.4.1 Scopo

La progettazione è il processo che trasforma i requisiti in un architettura che descriva la struttura del software e ne individui le componenti, allo scopo di fornire la base alle successive attività di realizzazione

2.2.4.2 Descrizione

L'architettura software prodotta in fase di progettazione deve soddisfare tutti i requisiti individuati durante il processo di analisi e descrive a tutti gli stakeholder la decomposizione del sistema in componenti, l'organizzazione di tali componenti, le interfacce necessarie all'interazione con l'ambiente e i paradigmi di composizione delle componenti. Per fare questo è bene fare uso di pattern architetturali

2.2.4.3 UML

La fase di progettazione deve essere supportata dai seguenti diagrammi realizzati in UML2.4:

- **Diagrammi delle classi** per descrivere la struttura e la gerarchia delle classi
- **Diagrammi di sequenza** per descrivere una determinata sequenza di azioni
- **Diagrammi di attività** per descrivere le attività necessarie per realizzare una data funzionalità
- **Diagrammi dei package** per descrivere l'architettura generale

2.2.5 Codifica

Le norme di progetto saranno ampliate con questa sezione non appena verranno concordate dal gruppo successivamente alla fase di progettazione.

3 Processi di Supporto

3.1 Scopo

Definisce norme per lo sviluppo e il mantenimento della documentazione prodotta durante il ciclo di vita del software. Inoltre definisce metodi per il controllo della qualità, di verifica e validazione di tali documenti.

3.2 Descrizione

Verranno definite norme riguardanti:

- utilizzo e l'accesso ai documenti;
- templates da utilizzare, l'uniformità di linguaggio, le convenzioni stilistiche e tipografiche;
- metodi di verifica e approvazione dei documenti;

3.3 Documentazione

La readazione di tutti i documenti viene assegnata dal *Project Manager_G* ai responsabili della stesura di tale documento secondo i ruoli_G definiti nella sezione 4.1.3. Successivamente se i redattori ritengono che il documento sia completo, su conferma da parte del *Project Manager_G*, il documento deve essere preso in visione da un *Verificatore_G*. Se il *Verificatore_G* ritiene che documento rispetta i requisiti imposti dal *Programmatore_G* e supera il controllo di qualità può proporre al *Project Manager_G* l'approvazione di tale documento. Se il documento viene approvato è giunto alla fase finale.

Tutti documenti essi dovranno a essere denominati secondo il seguente formalismo:

$$\text{NomeDocumento_}vX.Y.Z.pdf$$

dove *NomeDocumento* rappresenta il nome del documento e *vX.Y.Z* la versione come spiegata nella sezione 3.3.1. Il documento non può contenere lettere accentate. Nel caso il nome fosse composto da più parole, la prima lettera di ogni parola deve essere maiuscola e non devono esserci spazi, *underscore* o trattini a separare le parole.

Ogniqualevolta sia necessaria la citazione di una versione specifica di un documento, essa deve comprendere sia il nome che il numero di versione aderente al formato:

$$\text{Nome Documento } vX.Y.Z$$

3.3.1 Versione

La documentazione prodotta deve essere corredata del numero di versione attuale tramite la seguente codifica *vX.Y.Z* dove:

- **X** indica il numero crescente di uscite formali del documento. All'inizio di ogni fase il *Project Manager_G* deve cambiare tale indice seguendo la numerazione progressiva indicata e impostare a 0 gli indici Y e Z. L'indice deve seguire la numerazione progressiva indicata e non sono ammessi indici diversi da quelli elencati:

1. fase che si conclude con la *Revisione dei Requisiti*_G;
 2. fase che si conclude con la *Revisione di Progettazione*_G;
 3. fase che si conclude con la *Revisione di Qualifica*_G;
 4. fase che si conclude con la *Revisione di Accettazione*_G;
- **Y** indica la fase in cui il documento si trova. Nel momento in cui inizia l'attività di stesura il redattore del documento deve controllare che tale indice sia correttamente impostato a 0. All'inizio della verifica il *Verificatore*_G deve variare l'indice impostandolo a 1, dopo aver ricevuto il consenso dal *Project Manager*_G. Conclusa la verifica, il *Project Manager*_G provvede all'approvazione del documento e deve impostare l'indice a 2. L'indice deve seguire la numerazione progressiva indicata e non sono ammessi indici diversi da quelli elencati:
 0. stesura del documento;
 1. verifica del documento;
 2. approvazione del documento.
 - **Z** indica il numero crescente di modifiche minori. Ad ogni modifica effettuata deve corrispondere ad un'aggiunta di una voce nel diario delle modifiche, il redattore o il *Verificatore*_G devono aggiornare l'indice seguendo una numerazione progressiva. Non viene fissato un limite superiore per tale indice.

3.3.2 Stato

I documenti sono provvisti di uno stato_G. Tale stato può essere informale_G o formale_G.

3.3.2.1 Informale

Tutti i documenti saranno da ritenersi informali fino all'approvazione del *Project Manager*_G, il quale potrà richiederne una revisione ulteriore. L'utilizzo dei documenti informali è da considerarsi esclusivamente interno al gruppo e localizzato durante la fase di redazione e verifica di tali documenti.

3.3.2.2 Formale

I documenti approvati dal *Project Manager*_G si riterranno formali e pronti per essere distribuiti. Solo i documenti formali potranno essere distribuiti alla loro lista di distribuzione. Ogniqualevolta un documento formale venga modificato o rivisitato, la nuova versione è da considerarsi non formale fino ad approvazione del *Project Manager*_G quindi sarà trattata come un documento informale.

3.3.3 Uso

I documenti possono avere diverse liste di distribuzione ma una devono essere classificati in due principali categorie: uso interno ed esterno.

3.3.3.1 Interno

I documenti definiti ad uso interno non devono essere distribuiti all'esterno del gruppo di lavoro.

3.3.3.2 Esterno

I documenti definiti ad uso esterno devono essere distribuiti secondo i criteri della lista di distribuzione solamente se sono stati approvati dal *Project Manager*_G e se sono in stato formale_G.

3.3.4 Struttura dei documenti

I documenti devono rispettare una struttura prefissata. Per agevolare la redazione della documentazione è stato creato un *template* L^AT_EX contenente tutte le impostazioni stilistiche e grafiche citate in questo documento. Tale modello si può trovare nella repository_G in doc/template.

3.3.4.1 Prima pagina

Ogni documento deve avere una prima pagina che contiene le seguenti informazioni sul documento:

- nome del gruppo;
- nome del progetto;
- logo del gruppo;
- titolo del documento;
- versione del documento;
- stato del documento (informale_G o formale_G);
- uso del documento (interno_G o esterno_G);
- data di creazione del documento;
- data di ultima modifica del documento;
- nome e cognome dei redattori del documento;
- nome e cognome dei verificatori del documento;
- nome e cognome del responsabile che approva il documento;
- lista di distribuzione del documento;

3.3.4.2 Diario delle modifiche

Il diario delle modifiche del documento segue la prima pagina. La tabella deve essere ordinata per data in ordine decrescente, in modo che la prima riga corrisponda alla versione attuale del documento. Ogni riga del diario delle modifiche deve contenere:

- versione del documento dopo la modifica;
- nome e cognome dell'autore della modifica;
- data della modifica;

- titolo del documento;
- una breve descrizione delle modifiche svolte.

3.3.4.3 Indice

Al diario delle modifiche deve sempre seguire un indice delle sezioni e sottosezioni del documento.

3.3.4.4 Elenco tabelle e figure

All'indice delle sezioni e sottosezioni può seguire un elenco delle tabelle e delle figure. Nel caso non siano presenti figure o tabelle i rispettivi indici devono essere omessi.

3.3.4.5 Introduzione

Ogni documento deve essere provvisto di un'introduzione che ne spiega brevemente il contenuto. L'introduzione deve contenere due sottosezioni denominate *Scopo* e *Descrizione*. Lo scopo definisce a che proposito viene scritto il documento mentre la descrizione ne descrive brevemente gli argomenti trattati.

3.3.4.6 Sezioni

I documenti devono avere altre sezioni in modo da organizzare meglio i contenuti. Ogni sezione deve avere due sottosezioni denominate *Scopo* e *Descrizione*. Lo scopo definisce a che proposito viene scritta la sezione mentre la descrizione ne descrive brevemente gli argomenti trattati. Le sezioni possono inoltre avere altre sottosezioni. Queste sottosezioni possono avere al loro interno gerarchie annidate di sotto-sottosezioni, paragrafi e sotto-paragrafi. Questi non necessitano di *Scopo* e *Descrizione*.

3.3.4.7 Formattazione delle pagine

L'intestazione di ogni pagina, apparte la prima, deve contenere:

- logo del gruppo;
- nome del documento.

Il piè di pagina deve contenere:

- versione del documento;
- nome dell'università e anno accademico corrente;
- pagina corrente nel formato N di T dove N è il numero di pagina corrente e T è il numero di pagine totali;
- email del gruppo;
- licenza di distribuzione del documento.

3.3.5 Norme tipografiche

3.3.5.1 Stile del testo

- **Corsivo:** deve essere usato solo per indicare termini in lingua inglese, standard, citazioni, ruoli_G e nomi di documenti
- **Grassetto:** deve essere usato solo per parole significative che devono essere messe in risalto. Il grassetto deve essere usato con estrema cautela poiché può appesantire la lettura. Va applicato ai titoli e agli elementi che costituiscono un elenco se sono seguiti da una descrizione
- **Monospace:** deve essere usato solo per indicare i percorsi e i nomi dei file, codice e comandi
- **Maiuscolo:** deve essere usato solo per gli acronimi

3.3.5.2 Composizione

- **Elenchi:** ogni punto dell'elenco deve terminare con un punto e virgola, tranne l'ultimo che deve terminare con un punto. Gli elenchi devono essere puntati se non è necessario che gli elementi siano in un ordine preciso. Se l'elenco è una lista di elementi seguiti da descrizione devono terminare senza punteggiatura
- **Membri del gruppo:** i nomi dei membri del gruppo si devono indicare con nome e poi cognome. Sono state definite macro in \LaTeX con tutti i nomi del gruppo per facilitare la composizione dei documenti:
 - **Alessio Vitella:** $\backslash\text{AVI}$
 - **Andrea Venier:** $\backslash\text{AVE}$
 - **Nicola De Cao:** $\backslash\text{NDC}$
 - **Igor Baylyak:** $\backslash\text{IB}$
 - **Walter Sandon:** $\backslash\text{WS}$
 - **Thomas Pigarelli:** $\backslash\text{TP}$
 - **Anna Bonaldo:** $\backslash\text{AB}$
- **Ruoli:** i nomi dei ruoli devono essere riportati in corsivo e con la prima lettera maiuscola. Sono state definite macro in \LaTeX con tutti i ruoli definiti per facilitare la composizione dei documenti:
 - **Amministratore:** $\backslash\text{AM}$
 - **Analista:** $\backslash\text{AN}$
 - **Progettista:** $\backslash\text{PG}$
 - **Programmatore:** $\backslash\text{PR}$
 - **Verificatore:** $\backslash\text{VR}$
 - **Project Manager:** $\backslash\text{PM}$

- **Glossario:** ogni termine che possa causare ambiguità deve essere marcato da una lettera G scritta a pedice e riportata nel *Glossario_G*. È stata fatta una definite macro in L^AT_EX per facilitare la composizione dei documenti: `\gloss`
- **Note:** le note a piè di pagina possono contenere link o URL e devono essere indicate con il comando L^AT_EX: `\footnote`
- **L^AT_EX:** i riferimenti a L^AT_EX devono essere fatti utilizzando il comando L^AT_EX: `\LaTeX`
- **Revisioni:** i riferimenti alle revisioni devono essere fatti utilizzando la rispettiva macro che garantisce la corretta scrittura con la prima lettera di ogni parola che non sia una preposizione maiuscola :
 - **Revisione dei Requisiti:** `\RR`
 - **Revisione di Progettazione:** `\RP`
 - **Revisione di Qualifica:** `\RQ`
 - **Revisione di Accettazione:** `\RA`
- **Documenti:** i riferimenti ai documenti devono essere fatti utilizzando la specifica come definita nella sezione 3.3. Sono state definite macro L^AT_EX per i documenti in modo da facilitare la redazione dei documenti:
 - **Norme di Progetto:** `\NdP`
 - **Studio di Fattibilità:** `\SdF`
 - **Analisi dei Requisiti:** `\AdR`
 - **Piano di Progetto:** `\PdP`
 - **Piano di Qualifica:** `\PdQ`
 - **Glossario:** `\G10`

3.3.5.3 Formati

- **Nome files:** per i nomi dei file deve essere utilizzato il comando L^AT_EX: `\fileName`
- **Percorsi:** per i percorsi dei file deve essere utilizzato il comando L^AT_EX: `\filePath`
- **URLs e Links:** per gli indirizzi web completi deve essere utilizzato il comando L^AT_EX: `\url`, invece per i link deve essere utilizzato il comando L^AT_EX: `\href`
- **Date e Orari:** le date e gli orari devono seguire la notazione definita dallo standard *ISO 8601*¹: YYYY-MM-DD per le date e HH:MM per gli orari
- **Quantità e Unità di misura:** le quantità e le unità di misura dovranno aderire allo standard *ISO 80000*²

¹<http://www.iso.org/iso/iso8601>

²[TODO: link a iso 80000](#)

3.3.5.4 Tabelle

Ogni tabella presente all'interno dei documenti dev'essere accompagnata da una didascalia, in cui deve comparire un numero identificativo incrementale per la tracciabilità della stessa all'interno del documento.

3.3.5.5 Immagini e grafici

Le immagini e i grafici presenti all'interno dei documenti devono essere accompagnate da una didascalia, in cui deve comparire un numero identificativo incrementale per la tracciabilità della stessa all'interno del documento. Le immagini e i grafici devono essere nel formato *Scalable Vector Graphics* (SVG_G), così in questo modo si garantisce una maggior qualità dell'immagine in caso di ridimensionamento. Per consentire l'inclusione delle immagini e dei grafici nei documenti, essi dovranno essere convertiti nel formato PDF_G. Qualora non sia possibile salvare le immagine e i grafici in formato vettoriale è preferito il formato Portable Network Graphics (PNG_G).

3.4 Linguaggio

Nei documenti alcuni termini devono essere interpretati in modo analogo ai termini inglesi descritti in RFC 2119³:

- i termini *deve*, *è richiesto* e sinonimi stretti sono da intendersi con lo stesso significato di *MUST*;
- i termini *non deve*, *è richiesto che non* e sinonimi stretti sono da intendersi con lo stesso significato di *MUST NOT*;
- i termini *dovrebbe*, *si raccomanda*, *è preferibile* e sinonimi stretti sono da intendersi con lo stesso significato di *SHOULD*
- i termini *non dovrebbe*, *si raccomanda di non*, *è preferibile che non* e sinonimi stretti sono da intendersi con lo stesso significato di *SHOULD NOT*;
- i termini *può*, *opzionalmente* e sinonimi stretti sono da intendersi con lo stesso significato di *MAY*.

3.5 Configurazione

Il *Project Manager*_G, come descritto in 4.1.6, assegna ad ogni software utilizzato un referente che ne gestisce la configurazione, il mantenimento e la gestione dei problemi (spiegata in dettaglio nella sezione 3.8).

3.5.1 Documentazione

I documenti verranno distribuiti in formato PDF_G. Per la stesura dei documenti verrà utilizzato L^AT_EX come *word processor* e come linguaggio di markup. Verranno utilizzati i seguenti software per la produzione di materiale per i documenti:

³<http://www.ietf.org/rfc/rfc2119.txt>

- **Texmaker**⁴: (referente Nicola De Cao) verrà utilizzato come software da usare sia come editor che come processore $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$;
- **Dia**⁵: (referente **TODO:**) verrà utilizzato come software per creare diagrammi di flusso;
- **ArgoUML**⁶: (referente **TODO:**) verrà utilizzato come software per creare tutti i diagrammi UML_G ;

Tutto il software utilizzato è gratuito e *cross-platform*.

3.6 Verifica

Il processo di verifica produce i seguenti risultati:

- l'individuazione e l'applicazione di una strategia di verifica;
- individuazione e catalogazione dei difetti;
- identificazione di criteri per la verifica;
- stesura di una sezione relativa ai risultati della verifica.

La verifica di processi, documenti e prodotti deve essere eseguita durante l'intero sviluppo del progetto diattico, ed è a supporto degli altri processi e attività. Di conseguenza, servono modalità operative chiare e dettagliate per il *Verificatore_G*, che permettano di uniformare le attività di verifica svolte ed ottenere il miglior risultato possibile. Per applicare il processo di verifica il gruppo adotta diverse tecniche che verranno descritte di seguito.

3.6.1 Tecniche di analisi

- **Analisi statica:** tale tecnica è attuabile sia alla documentazione sia al codice, consiste nell'individuazione di errori ed anomalie ad esempio effettuando una lettura critica del testo a largo spettro oppure più mirata. Il *Verificatore_G* controllerà i documenti e il codice utilizzando le seguenti tecniche:
 - **Walkthrough:** consiste in una lettura a largo spettro del documento/codice in questione che permetta di individuare i possibili errori o anomalie presenti. Con questa tecnica non si conosce la tipologia o l'insieme di errori da ricercare, bensì ogni aspetto è attentamente analizzato al fine di ricavare, successivamente, delle categorie di errori tipici. Ogni difetto rilevato è discusso con gli autori allo scopo di evitare incomprensioni e concordare le modifiche necessarie. Il *walkthrough_G* è indispensabile nelle prime fasi di sviluppo quando non si possiede ancora una chiara visione dei possibili errori. Usando più volte questa tecnica è possibile stilare una lista di controllo dove vengono archiviati gli errori più spesso rilevati. Man mano che il *Verificatore_G* rende più esaustiva la lista di controllo, si ricorre sempre meno all'utilizzo della tecnica di *walkthrough_G* a favore della tecnica di *inspection_G*;

⁴<http://www.xm1math.net/texmaker/>

⁵<https://wiki.gnome.org/Apps/Dia/>

⁶<http://argouml.tigris.org/>

- **Inspection:** l'inspection_G si basa sulla lettura mirata dei documenti/codice. Durante tale lettura si cercano gli errori segnalati nella lista di controllo. Progressivamente, con l'acquisizione di esperienza e grazie al precedente uso della tecnica di walkthrough_G, la lista di controllo viene estesa e specializzata, rendendo l'inspection_G sempre più efficace;

3.6.2 Classificazione dei problemi

A corredo della Lista di controllo **TODO:** , vengono classificate le categorie delle problematiche più comuni e le giuste tecniche di risoluzione, adattate alla gravità dei relativi errori. Essi si dividono per ambito e si ordinano secondo una priorità dettata dall'urgenza di risoluzione. Ambiti:

- Processi;
- Documentazione;
- Codice.

Urgenza:

- **Alta:** l'errore deve essere risolto non appena possibile. In ambito di codifica, la gravità è tale da poter coinvolgere più unità e richiedere di ripetere numerosi test. Una mancata gestione del problema può determinare difetti nel prodotto così come un ritardo nella risoluzione può comportare considerevoli aumenti nei tempi e nei costi dei singoli processi;
- **Media:** l'errore deve essere risolto entro una scadenza stabilita considerando la presenza di altri errori di pari urgenza. In media la risoluzione non dovrebbe richiedere più di 5 giorni lavorativi;
- **Bassa:** l'errore è di bassa entità e la sua risoluzione non richiede intervento tempestivo. La scadenza per la sua risoluzione coincide con la milestone_G successiva.

Metodo di gestione del problema:

- **Ticket:** il Verificatore apre un ticket_G su Asana_G per la risoluzione del problema riscontrato (dettagli nella sezione **TODO:**);
- **Aggiunta alla checklist_G:** il Verificatore_G scorre la Lista di controllo per verificare che sia presente un riferimento alla famiglia di errori di cui fa parte l'errore rilevato; in caso negativo aggiunge la nuova famiglia di errori;
- **Correzione immediata:** il Verificatore_G provvede autonomamente alla correzione dell'errore, segnalando la modifica effettuata tramite sistema di versionamento_G, indicando un messaggio di commit_G esaustivo (per approfondimenti si rimanda alla sezione sulle norme di versionamento_G **TODO:**).

Ambito	Problema	Urgenza	Azione
Processi	Ritardo di 3 giorni	Alta	Ticket
	Errore di progettazione	Alta	Ticket
	Errore in requisiti e use case	Alta	Ticket
Documentazione	Errore di compilazione	Alta	Ticket
	Errore concettuale	Alta	Ticket
	Errore in tabelle o immagini	Media	Ticket
	Errore ortografico	Media	Correzione immediata
	Termine non in glossario	Bassa	Ticket
	Norma stilistica non rispettata	Bassa	Correzione immediata
	Errore di formattazione	Bassa	Ticket

Tabella 2: Classificazione dei problemi

3.6.3 Diagrammi UML

Al *Verificatore_G* è richiesto il controllo dei diagrammi UML prodotti:

- **Diagrammi di caso d'uso:** Il controllo dei diagrammi di caso d'uso deve avvenire manualmente, controllando il rispetto delle specifiche UML_G 2.4 e il corretto uso delle relazioni di inclusione ed estensione. Il diagramma di caso d'uso deve rappresentare fedelmente quanto descritto dal caso d'uso.

3.6.4 Documentazione

Il processo di verifica viene istanziato nel momento in cui l'output di un processo passa dalla versione X.0.Z alla versione X.1.0. È compito del responsabile del documento notificare i *Verificatori* dell'inizio dell'attività di verifica. Attraverso il diario delle modifiche è possibile focalizzare l'attenzione maggiormente sulle sezioni che hanno subito dei cambiamenti dall'ultima verifica, riducendo il tempo necessario al controllo. Per eseguire un'accurata verifica dei documenti redatti è necessario seguire il seguente protocollo:

- **Controllo sintattico e del periodo:** Utilizzando TeXstudio e GNU Aspell vengono evidenziati e corretti gli errori di grammatica più evidenti. Gli errori di sintassi, di sostituzione di lettere che provocano la creazione di parole grammaticalmente corrette ma sbagliate nel contesto ed i periodi di difficile comprensione necessitano dell'intervento di un verificatore umano. Per questa ragione ciascun documento dovrà essere sottoposto ad una attenta lettura da parte dei verificatori per individuare tali errori
- **Rispetto delle norme di progetto:** Sono state definite norme tipografiche di carattere generale in *Norme di Progetto_G*. Esse impongono una struttura

dei documenti che non può essere verificata in maniera automatica. La verifica delle norme per cui non è stato definito uno strumento automatico richiede che i Verificatori eseguano un controllo sul rispetto di quelle norme in ciascun documento

- **Verifica delle proprietà di glossario:** Nella redazione del *Glossario_G* è possibile automatizzare tale verifica. Utilizzando l'apposito script, vengono individuate tutte le segnalazioni di una parola del glossario in un documento in maniera automatica, ed applicato l'opportuno stile. Tale script è descritto nella sezione TODO: Il Verificatore dovrà quindi controllare che l'output finale sia conforme a quanto precisato relativamente al *Glossario_G*
- **Miglioramento del processo di verifica:** quando i Verificatori eseguono *walk-through* di un documento, dovranno riportare gli errori più frequentemente trovati. Grazie a tale pratica sarà possibile eseguire *inspection* su tali errori nelle verifiche future
- Norme stilistiche TODO:
- **Segnalazione degli errori riscontrati:** Attenendosi a quanto riportato nelle tabelle 3 e 2 il Verificatore deve generare ticket secondo quanto descritto nella sezione TODO:

3.7 Validazione

3.8 Risoluzione dei Problemi

Dal *Project Manager_G* viene aperto un task (per approfondire vedere sezione TODO:) con scadenza indefinita con titolo:

Nome del Software - Risoluzione Problemi

Questo task viene assegnato al referente di quel determinato software. Il task indica che tale membro del team deve provvedere a risolvere ogni problema riscontrato nell'utilizzo del software dagli altri membri del team.

4 Processi di Organizzazione

Tali processi eseguono funzioni a livello organizzativo aziendale, per supportare altri processi primari, di supporto o organizzativi. I processi organizzativi aiutano nel definire, controllare e migliorare gli altri processi.

4.1 Processi di Management

4.1.1 Scopo

I processi di management contengono le attività generiche e i task, che possono essere impiegati da un qualsiasi gruppo che deve gestire i suoi rispettivi processi. Il *Project Manager_G* responsabile per la gestione del prodotto, la gestione del progetto, la gestione dei *tasks_G* inerenti ai processi come quelli di acquisizione, fornitura, sviluppo, manutenzione o i processi di supporto.

4.1.2 Descrizione

Verranno definite norme riguardanti:

- Catalogazione dei ruoli e i loro compiti specifici;
- L'identificazione delle modalità di comunicazione interne ed esterne al gruppo;
- Definizione di metodologie di Esecuzione e Controllo.

4.1.3 Ruoli

Durante lo sviluppo del progetto vi saranno diversi ruoli che i membri del gruppo andranno a ricoprire. Tali ruoli rappresentano figure aziendali specializzate, indispensabili per il buon esito del progetto. Ciascun componente del gruppo dovrà ricoprire almeno una volta ogni ruolo. Si deve inoltre certificare che non vi siano conflitti di interesse nello svolgimento delle attività di verifica e di approvazione. Per garantire che la rotazione dei ruoli non provochi conflitti è necessario che le attività di stesura e verifica vengano pianificate dettagliatamente e che i soggetti interessati rispettino i compiti a loro assegnati. Sarà poi compito del *Verificatore_G* controllare attentamente il diario delle modifiche di ogni documento per individuare eventuali incongruenze. Si descrivono ora i diversi ruoli di progetto, con le relative responsabilità e le modalità operative affinché essi possano svolgere i compiti assegnati con l'ausilio dei software scelti per il progetto.

4.1.3.1 Project Manager

Il *Project Manager_G* rappresenta il progetto, in quanto accentra su di sé le responsabilità di scelta ed approvazione, ed il gruppo, in quanto presenta al committente i risultati del lavoro svolto. Detiene il potere decisionale, quindi la responsabilità su:

- Pianificazione, coordinamento e controllo delle attività;
- Gestione e controllo delle risorse;
- Analisi e gestione dei rischi;

- Approvazione dei documenti;
- Approvazione dell'offerta.

Di conseguenza, ha il compito di assicurarsi che le attività di verifica vengano svolte sistematicamente seguendo le Norme di Progetto, vengano rispettati i ruoli e le competenze assegnate nel *Piano di Progetto_G*, non vi siano conflitti di interesse tra *Redattori_G* e *Verificatori_G*. Egli è l'unico a poter decidere l'approvazione di un documento e a sancirne la distribuzione. Ha inoltre l'incarico di gestire la creazione e l'assegnazione dei *ticket_G* delle macro-fasi e di assegnare ad un membro del gruppo il ruolo di responsabile di quest'ultima. Redige il *Piano di Progetto_G* e collabora alla stesura del *Piano di Qualifica_G*, in particolare nella sezione relativa alla pianificazione.

4.1.3.2 Amministratore

L'*Amministratore_G* è responsabile del controllo, dell'efficienza e dell'operatività dell'ambiente di lavoro. Le mansioni di primaria importanza che gli competono sono:

- Ricerca di strumenti che possano automatizzare qualsiasi compito che possa essere tolto all'umano;
- Risoluzione dei problemi legati alle difficoltà di gestione e controllo dei processi e delle risorse. La risoluzione di tali problemi richiede l'adozione di strumenti adatti;
- Controllo delle versioni e delle configurazioni del prodotto;
- Gestione dell'archiviazione e del versionamento della documentazione di progetto;
- Fornire procedure e strumenti per il monitoraggio e la segnalazione per il controllo qualità.

Redige le *Norme di Progetto_G*, dove spiega e norma l'utilizzo degli strumenti, redige la sezione del *Piano di Qualifica_G* dove vengono descritti strumenti e metodi di verifica.

4.1.3.3 Analista

L'*Analista_G* è responsabile delle attività di analisi. Le responsabilità di spicco per tale ruolo sono:

- Produrre una specifica di progetto comprensibile, sia per il Proponente_G, sia per il Committente_G che per il Progettista_G, e motivata in ogni suo punto;
- Comprendere appieno la natura e la complessità del problema.

Redige lo *Studio di Fattibilità_G*, l'*Analisi dei Requisiti_G* e parte del *Piano di Qualifica_G*. Partecipa alla redazione del *Piano di Qualifica_G* in quanto conosce l'ambito del progetto ed ha chiari i livelli di qualità richiesta e le procedure da applicare per ottenerla.

4.1.3.4 Progettista

Il *Progettista_G* è responsabile delle attività di progettazione. Le responsabilità di tale ruolo sono:

- Produrre una soluzione attuabile, comprensibile e motivata;
- Effettuare scelte su aspetti progettuali che applichino al prodotto soluzioni note ed ottimizzate;
- Effettuare scelte su aspetti progettuali e tecnologici che rendano il prodotto facilmente mantenibile.

Redige la *Specifica Tecnica*, la *Definizione di Prodotto* e le sezioni inerenti le metriche di verifica della programmazione del *Piano di Qualifica_G*.

4.1.3.5 Programmatore

Il *Programmatore_G* è responsabile delle attività di codifica e delle componenti di ausilio necessarie per l'esecuzione delle prove di verifica e validazione. Le responsabilità di tale ruolo sono:

- Implementare rigorosamente le soluzioni descritte dal *Progettista_G*, da cui seguirà quindi la realizzazione del prodotto;
- Scrivere codice documentato, versionato, mantenibile e che rispetti gli standard stabiliti per la scrittura del codice;
- Implementare i test sul codice scritto, necessari per prove di verifica e validazione.

Redige il *Manuale Utente* e produce una abbondante documentazione del codice.

4.1.3.6 Verificatore

Il *Verificatore_G* è responsabile delle attività di verifica. Ha il compito di effettuare la verifica dei documenti utilizzando gli strumenti e i metodi proposti dal *Piano di Qualifica_G* e attenendosi a quanto descritto nelle *Norme di Progetto_G*. Le responsabilità di tale ruolo sono:

- Implementare rigorosamente le soluzioni descritte dal *Progettista_G*, da cui seguirà quindi la realizzazione del prodotto;
- Assicurare che l'attuazione delle attività sia conforme alle norme stabilite;
- Controllare la conformità di ogni stadio del ciclo di vita del prodotto.

Redige la sezione del *Piano di Qualifica_G* che illustra l'esito e la completezza delle verifiche e delle prove effettuate.

4.1.4 Comunicazione

4.1.4.1 Esterna

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

starware.swe@gmail.com

Tale indirizzo deve essere l'unico canale di comunicazione esistente tra il gruppo di lavoro e l'esterno. Solo il *Project Manager_G* ha accesso all'indirizzo ed è quindi l'unico a poter comunicare con il committente del progetto. È compito del *Project Manager_G* informare i membri del gruppo delle discussioni avvenute.

4.1.4.2 Interna

Per le comunicazioni interne informali è stato scelto l'uso di *Telegram_G*, per il suo utilizzo tramite *browser_G* che permette, quindi, un accesso alle conversazioni sia da pc che da *smartphone_G*. Per lo scambio e la condivisione di file e appunti è invece utilizzato il servizio cloud di *Google Drive_G*. Inoltre è permesso l'uso di *Skype_G* per effettuare conferenze di gruppo o anche tra singoli membri di esso.

4.1.5 Riunioni

4.1.5.1 Frequenza

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

4.1.5.2 Convocazione riunione

Il *Project Manager_G* ha il compito di convocare le riunioni generali, ovvero le riunioni in cui vengono convocati tutti i membri del gruppo. Qualora ve ne sia la necessità, qualsiasi componente può richiedere la convocazione di una riunione, ma tale richiesta deve essere inoltrata al *Project Manager_G* che a cui spetta il compito di vagliarla. È inoltre possibile e auspicabile che possano essere necessarie riunioni tra specifici membri: ad esempio, nella fase di progettazione può essere utile la collaborazione tra Progettista e Analista, senza richiedere la presenza di persone non direttamente coinvolte, che verranno comunque informate delle decisioni prese tramite verbale. Il *Project Manager_G* deve convocare la riunione attraverso la creazione di un *task_G* su *Asana_G*, assegnato a tutti i membri del gruppo, avente le seguenti caratteristiche:

- **Titolo:** convocazione riunione n. X, dove X indica il numero crescente di riunioni effettuate;
- **Data:** la data del task dovrà essere impostata alla data prevista per la riunione;
- **Ordine del Giorno:** elenco ordinato delle varie voci da esaminare sotto forma di commento al task.

Ogni componente del gruppo deve confermare la propria presenza al *Project Manager_G* il prima possibile. In caso di mancata risposta o di immotivata assenza il *Project Manager_G* ha il dovere di contattare anche telefonicamente colui che non ha fornito una risposta in tempo utile. Il *Project Manager_G*, una volta ricevute le risposte e

verificata la presenza o assenza dei membri richiesti, può decidere se confermare, annullare o spostare la riunione, per permettere la partecipazione agli eventuali assenti. La conferma, l'annullamento e lo spostamento dell'assemblea devono essere effettuati attraverso modifiche o cancellazione del task_G su Asana_G .

4.1.5.3 Verbale

All'apertura della riunione, verificata la presenza dei membri previsti, viene scelto un segretario che avrà il compito di annotare ogni argomento trattato e di redigere il verbale dell'assemblea. Il segretario deve inoltre controllare che venga seguito l'ordine del giorno in modo da non tralasciare alcun punto.

Riunione interna Il verbale di riunione interna è un documento interno informale che consente di tracciare gli argomenti discussi durante la riunione. Verrà redatto dal segretario della riunione, ruolo scelto di volta in volta a rotazione tra i presenti. Si consiglia di redigere il verbale di una riunione interna in un documento di testo da allegare come commento al task_G creato per la riunione su Asana_G , in modo da rendere sempre disponibile il contenuto dello stesso.

Riunione esterna In caso di riunione con il committente od il proponente, il verbale è un documento ufficiale che può avere valore normativo e quindi deve essere redatto seguendo criteri specifici. Per agevolare la scrittura di tale documento è stato preparato un *template* \LaTeX che ne definisce la struttura e ne organizza i contenuti. Vi è quindi l'obbligo di seguire il sopraccitato schema per creare e condividere il verbale.

4.1.6 Pianificazione

Il *Project Manager_G* ha la responsabilità di preparare il piano per l'esecuzione del processo, ponendo particolare attenzione a :

- Programmazione del lavoro per il completamento tempestivo dei task_G ;
- Stima dello sforzo necessario;
- Assegnazione delle risorse necessarie al completamento del task_G ;
- Assegnazione delle responsabilità;
- Quantificazione dei rischi associati ai task_G ;
- Misure di controllo qualità che devono essere impiegate durante il processo;
- Costo associato all'esecuzione del processo.

4.1.6.1 Pianificazione delle attività

Per pianificare le attività il *Project Manager_G* deve realizzare un diagramma di *Gantt_G* per ciascuna fase indicata nella sezione Pianificazione del *Piano di Progetto_G*. Per assolvere a questo compito è stato scelto il software open source *ProjectLibre_G* per i seguenti motivi:

- Permette di creare diagrammi di *Gantt_G* e di *Pert_G* in maniera facile ed intuitiva;

- Essendo opensource e basato sulla piattaforma Java, è compatibile con i sistemi operativi Linux, Mac OS e Windows. Permette quindi ad ogni componente di installarlo senza vincoli di piattaforma;

Sarà compito del *Project Manager_G* monitorare l'avanzamento relativo ad ogni fase del processo, tenendo in considerazione e risolvendo le eventuali dipendenze tra attività da eseguire.

4.1.6.2 Assegnazione dei task

Il *Project Manager_G*, una volta individuati e stimati i task da svolgere, ha la responsabilità di assegnarne l'esecuzione ai vari membri del gruppo. Per la suddivisione e l'assegnazione dei *task_G* è stato scelto di utilizzare l'applicazione web *Asana_G* per i seguenti motivi:

- Permette di creare e assegnare *task_G* in maniera facile ed intuitiva;
- Permette di monitorare chiaramente le scadenze dei *task_G*;
- Da la possibilità dividere i *task_G* in *subtask_G*;
- Esiste anche un applicazione per *smartphone_G* rendendo molto più accessibile l'utilizzo di *Asana_G*.

Sarà inoltre compito del *Project Manager_G* assegnare una deadline per il completamento del *task_G* e risolvere eventuali problemi legati a ritardi nel completamento degli stessi.

4.1.7 Esecuzione e controllo

4.2 Processi di gestione dell'infrastruttura

4.2.1 Scopo

Sono processi per stabilire e mantenere le infrastrutture necessarie a qualsiasi altro processo. Le infrastrutture possono includere HardWare, SoftWare, strumenti, tecniche, standards, strutture per lo sviluppo o manutenzione.

4.2.2 Descrizione

Verranno definite norme riguardanti:

- Definizione e documentazione sulle modalità di *ticketing_G* ;
- Definizione e documentazione sulle modalità di *versioning_G*;
- Definizione e documentazione sulla struttura del sistema di *repository_G*.

4.2.3 Ticketing

4.2.3.1 Struttura Ticket

I *ticket_G* creati devono avere le seguenti caratteristiche:

- **Titolo:** Rappresenta in maniera sintetica e significativa il *task_G*;
- **Assegnatari/o:** I *tasks_G* o *subtasks_G* devono sempre essere assegnati ad uno o più membri del gruppo;
- **Data:** Ogni *tasks_G* o *subtasks_G* deve sempre avere una data, decisa attraverso i diagrammi di *Gant_G* e *Pert_G*, entro il quale essere completato;
- **Commento:** Nel caso fossero necessarie delucidazioni al *task_G* bisogna chiarirle in maniera concisa sotto forma di commento;
- **Label:** Bisogna segnalare lo stato del *task_G* attraverso una etichetta appropriata;
- **Subtask:** In caso di necessità è possibile suddividere il *task_G* in diversi *subtask_G* imputabili ai membri del gruppo.

4.2.3.2 Etichette

Lo stato di ogni *ticket_G* deve essere identificato da un appropriata etichetta:

- **Aperto:** Etichetta riservata ai *ticket_G* non ancora assegnati;
- **Accettato:** Etichetta riservata ai *ticket_G* una volta in cui il membro del gruppo a cui sono imputati ne ha preso visione;
- **In Corso:** Etichetta riservata ai *ticket_G* in fase di elaborazione;
- **In Attesa:** Etichetta riservata ai *ticket_G* che non possono entrare in fase di elaborazione a causa di dipendenze con altri ticket non ancora completati;

- **Completato:** Etichetta riservata ai $ticket_G$ per cui la fase di elaborazione è stata completata con successo;
- **Verificato:** Etichetta riservata ai $ticket_G$ di cui è stata verificato l'effettivo completamento e l'assenza di errori;

4.2.3.3 Ciclo di vita del Ticket

Ogni $ticket_G$ rappresenta un attività_G o un $task_G$ e deve rispettare il seguente iter:

1. Creazione di un $task_G$;
2. Il *Project Manager*_G assegna il $task_G$ ad uno o più membri del gruppo, nel caso l'assegnatario non ne abbia ancora preso visione o non abbia ancora deciso se accettare il $task_G$, sarà necessario impostare come etichetta **Aperto**;
3. L'assegnatario può decidere se accettare o meno il $task_G$:
 - Accettazione: L'assegnatario dovrà modificare l'etichetta del $task_G$ in **Accettato**.
 - Rifiuto: L'assegnatario dovrà motivare il rifiuto del $task_G$ e, nel caso le motivazione fossero reputate pertinenti dal *Project Manager*_G, verrà ripetuto il punto 2.
4. Se il $task_G$ fosse dipendente dal completamento di un altro $task_G$ in stato di esecuzione, sarà necessario modificarne l'etichetta in **In attesa**;
5. Esecuzione del $task_G$;
6. Una volta completato il $task_G$ l'assegnatario dovrà modificarne l'etichetta in **Completato**;
7. Ogni tag completato dovrà essere vagliato attraverso procedure di verifica.

4.2.3.4 Ticket di Verifica

Per ogni $task_G$ marcato da etichetta **Completato** deve esserne verificata la correttezza e l'effettivo completamento attraverso la seguente procedura:

1. Al completamento di un $ticket_G$ il *Project Manager*_G deve creare un $task_G$ marcato dall'etichetta **Verificare** ed assegnarlo ad un *Verificatore*_G;
2. Il *Verificatore*_G, nel caso riscontri anomalie o errori deve creare i $task_G$ necessari alla correzione degli stessi. Nel caso dell'assenza di errori il *Verificatore*_G chiude il $task_G$ e ne modifica lo stato impostando l'etichetta **Verificato**.
3. Il *Project Manager*_G assegna gli eventuali $task_G$ creati dal *Verificatore*_G ponendo particolare attenzione a non far sorgere conflitti di interesse.

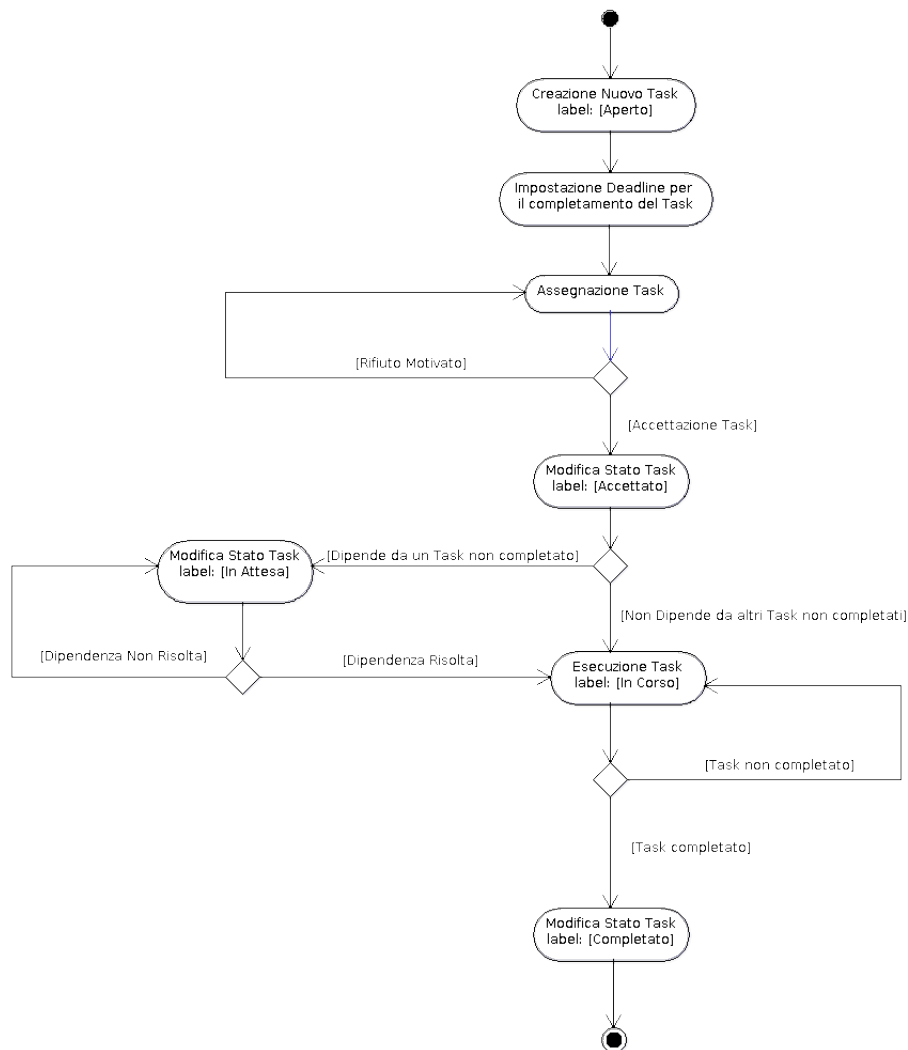


Figura 1: Diagramma di gestione dei Ticket.

4.2.4 Versioning

Come sistema di versionamento è stato scelto di usare Git_G. I motivi principali che ne hanno determinato la scelta sono:

- è gratuito e non proprietario
- è un repository distribuito con la possibilità di commit e revert locali
- è cross-platform
- possiede un'abbondante ma semplice documentazione
- è già stato usato da alcuni elementi del gruppo

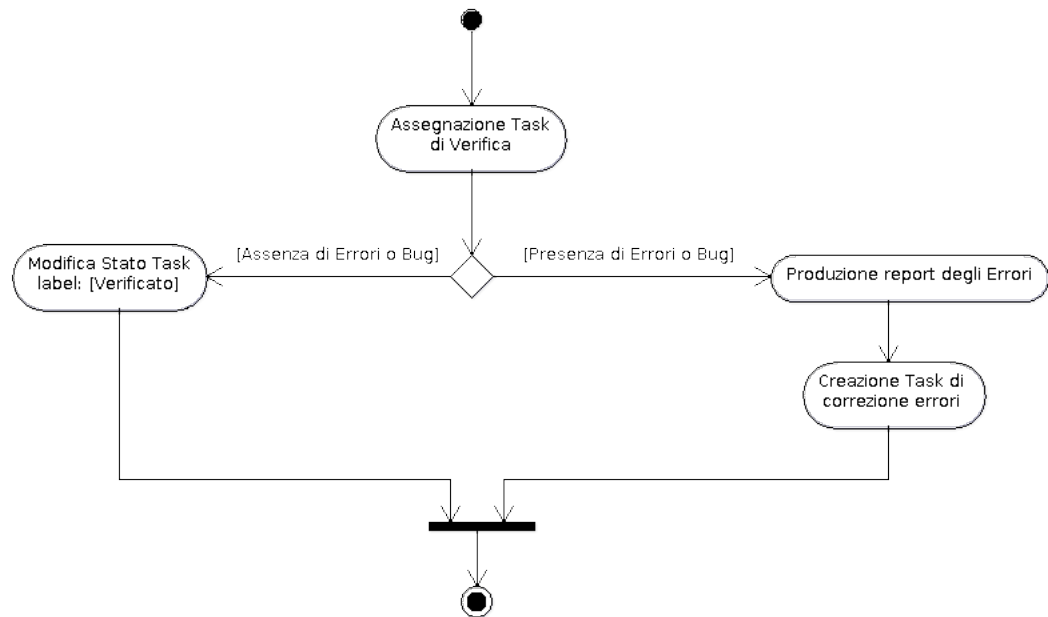


Figura 2: Diagramma di gestione dei Ticket di Verifica.

4.2.5 Repository

Viene creata una repository_G su github all'indirizzo <https://github.com/lordnikolai/starware.swe15>. Ogni membro deve clonare la repository in locale tramite il comando `git clone`.

4.2.5.1 Comandi

Commit Il comando per fare un commit è `git commit -a -m "<messaggio>".` Il commit va fatto solo in uno stato consistente dei documenti modificati, dove per consistente si intende che non vi siano porzioni di testo interrotte o codice non completo. Il messaggio da associare al commit è obbligatorio e deve esprimere in modo sintetico ma chiaro quale sia l'elemento in cui si è concentrato il commit e quale delle seguenti azioni sia stata fatta:

- Aggiunta di un nuovo elemento
- Correzione o modifica
- Miglioramento o ampliamento

Pull È preferibile fare un pull ogni volta che si comincia a lavorare da locale. Si raccomanda di controllare sempre il branch a cui si sta lavorando

Push Ogni volta che si ha un commit con informazioni che potrebbero interessare ad altri membri del gruppo bisogna fare una push della repository sulla directory online. Controllare che il commit rispetti le regole del paragrafo precedente.

4.2.5.2 Struttura albero delle directory

La repository è organizzata secondo il seguente albero delle directory

- doc
 - 1_RR
 - interni
 - esterni
 - 2_RP
 - interni
 - esterni
 - 3_RQ
 - interni
 - esterni
 - 4_RA
 - interni
 - esterni
 - template
- quizzipedia

La cartella doc contiene tutti i documenti, la cartella quizzipedia contiene il codice sorgente dell' applicazione (la struttura interna sarà specificata in fase di progettazione). All' interno di doc c'è la cartella template che contiene tutti i file per definire lo stile o creare un nuovo documento \LaTeX . Le 4 cartelle 1_RR, 2_RP, 3_RQ e 4_RA rappresentano rispettivamente i documenti per la revisione dei requisiti, revisione di progettazione, revisione di qualifica e revisione di accettazione. All' interno di ognuna di queste 4 directory ci sono altre 2 cartelle per dividere i documenti interni (*Norme di Progetto_{Ge} Studio di Fattibilità_G*) da quelli esterni (*Analisi dei Requisiti_G, Glossario_G, Piano di Progetto_{Ge} Piano di Qualifica_G*).