

16 gennaio 2016

StarWare

Quizzipedia: software per la gestione di questionari



Norme di Progetto

Informazioni sul documento

Nome Documento	Norme di Progetto
Versione	1.0.11
Stato	<i>In lavorazione</i>
Uso	<i>Interno</i>
Data Creazione	30 novembre 2015
Data Ultima Modifica	16 gennaio 2016
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
1.0.11	Nicola De Cao	2016-01-10	Finita parte Ticketing e modifiche minori
1.0.10	Nicola De Cao	2016-01-07	Sistemata organizzazione per processi
1.0.9	Andrea Venier	2016-01-03	Descrizione Software UML
1.0.8	Andrea Venier	2016-01-02	Modifiche Ticketing
1.0.7	Nicola De Cao	2015-12-24	Processi di supporto
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	Creazione documento

Tabella 1: Versionamento del documento

Indice

Elenco delle tabelle

Elenco delle figure

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 alla fine di questo documento. Ogni occorrenza di vocaboli presenti nel glossario è marcata da una G maiuscola in pedice (e.g. pedice_G) ed se cliccata porta direttamente alla definizione.

2 Processi primari

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 lo *Studio di fattibilità*, *Piano di qualifica*, e il *Piano di progetto*.

2.1.3 Studio di fattibilità

È compito degli *Analisti_G* redigere uno *Studio di fattibilità* in cui per ogni capitolato bisogna individuare:

- Aspetti positivi
- Fattori di rischio

Per il capitolato scelto lo *Studio di fattibilità* deve contenere anche una descrizione del dominio applicativo e del dominio tecnologico. Infine devono essere spiegati i motivi della scelta del gruppo.

2.1.4 Piano di Qualifica

Il *Piano di qualifica* 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.

2.1.5 Piano di Progetto

Il *Piano di progetto* 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 *Responsabile_G*, assieme all'*Amministratore_G*, deve presentare nel *Piano di progetto* diagrammi di Gantt_G 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 *Responsabile_G* e qualsiasi altro ruolo o tra *Verificatore_G* e qualsiasi altro ruolo.

2.1.5.1 Strumenti

LibreOffice Calc Per l'elaborazione dei dati si utilizza il software Calc del pacchetto LibreOffice in quanto tale prodotto è open-source_G. Calc viene utilizzato per elaborare i dati prodotti con ProjectLibre_G e produrre:

- grafici a torta per l'utilizzo delle risorse;
- grafici a torta per il costo dedicato a ciascuna risorsa;
- istogrammi per le ore assegnate ad ogni componente del gruppo;
- calcolo della metrica SV (Schedule variance_G);
- calcolo della metrica CV (Cost variance_G);
- tabelle per il confronto tra preventivo e consuntivo;
- istogrammi per il confronto tra ore preventivate e ore realmente impiegate da ciascuna risorsa.

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. attività di *Analisi_G*;

2. attività di *Progettazione_G*;
3. attività di *Codifica_G*.

2.2.3 Analisi dei requisiti

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

2.2.3.1 Struttura di un caso 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

2.2.3.2 Struttura di un requisito

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 struttura:

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, descrive 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*
- **Validazione:** verrà inserito un link al metodo deciso per validare il requisito

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.

2.2.3.3 Struttura di una fonte

Ogni fonte deve avere la seguente struttura:

- **Tipologia:** serve ad identificare se si tratta di una fonte esterna oppure se la fonte è un caso d'uso
- **Descrizione:** descrizione sintetica della fonte

2.2.3.4 Struttura di un metodo di validazione

Ogni metodo di validazione deve avere la seguente struttura:

- **Descrizione:** descrizione sintetica della metodo di validazione

2.2.4 Progettazione

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. 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_G 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

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

Questa sezione delle *Norme di progetto* sarà ampliata non appena verranno concordate dal gruppo norme riguardanti la *Progettazione_G* in previsione della *Revisione di progettazione_G*.

2.2.5 Validazione

Le *Norme di progetto* saranno ampliate con questa sezione non appena verranno concordate dal gruppo.

2.2.6 Codifica

Le *Norme di progetto* saranno ampliate con questa sezione non appena verranno concordate dal gruppo.

2.2.7 Strumenti

2.2.7.1 Tracciamento dei requisiti

Come software_G per il tracciamento dei requisiti si è scelto di utilizzare Tracy_G, realizzato dal gruppo *Don't Panic* durante l'anno accademico 2012/2013. Si tratta di un'applicazione web per la gestione dei requisiti e casi d'uso. Permette inoltre l'esportazione in formato L^AT_EX.

Aggiunta di un nuovo caso d'uso Per aggiungere un nuovo caso d'uso alla gerarchia dei casi d'uso è necessario eseguire seguenti passi:

1. eseguire l'accesso a tracy_G;
2. selezionare Use Cases nella sezione Analisi Requisiti;
3. selezionare Create Use Case dal menu Operations;
4. nel caso il caso d'uso possedesse un padre impostare Parent;
5. impostare la descrizione , la pre-condizione e la post-condizione del caso d'uso;
6. premere il pulsante Create per confermare la creazione del caso d'uso.

Aggiunta di un nuovo requisito Per aggiungere un nuovo requisito al progetto è necessario eseguire i seguenti passi:

1. eseguire l'accesso a tracy_G;
2. selezionare Requirements nella sezione Analisi Requisiti;
3. selezionare Create Requirement dal menu Operations;
4. impostare i valori per: Category, Priority e Apported;
5. nel caso il requisito possedesse un padre impostare Parent;
6. nel caso il requisito possedesse un padre fonte Sources;
7. impostare Description;
8. impostare un metodo di validazione in Validation;
9. premere il pulsante Create per confermare la creazione del requisito.

Aggiunta di un nuova fonte Per aggiungere una nuova fonte, che servirà per identificare da dove si è identificato un requisito, è necessario eseguire i seguenti passi:

1. eseguire l'accesso a `tracyG`;
2. selezionare Sources nella sezione Analisi Requisiti;
3. selezionare Create External Source dal menu Operations;
4. impostare Description;
5. premere il pulsante Create per confermare la creazione della fonte.

Aggiunta di una nuovo metodo di validazione Per aggiungere un nuovo metodo di validazione, che servirà per identificare come validare un requisito, è necessario eseguire i seguenti passi:

1. eseguire l'accesso a `tracyG`;
2. selezionare Requirements nella sezione Analisi Requisiti;
3. selezionare Manage Validation Methods dal menu Operations;
4. impostare Description;
5. premere il pulsante Create per confermare la creazione del nuovo metodo di validazione.

2.2.7.2 UML

Per la modellazione dei diagrammi dei caso d'uso, diagrammi di sequenza e diagrammi di attività è stato scelto l'editor `Draw.ioG`. Le caratteristiche che hanno fatto propendere il gruppo ad usare questo software_G sono state un interfaccia elementare ed intuitiva, il supporto di `UMLG2.0` e la possibilità di utilizzare questa applicazione sia da `browserG` che da `DesktopG`.

3 Processi di supporto

3.1 Scopo

Definisce norme per lo sviluppo e il mantenimento della documentazione, a supporto dei processi basi, 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 per i documenti, l'uniformità di linguaggio, le convenzioni stilistiche e tipografiche;
- metodi di verifica e approvazione del materiale prodotto;
- metodi di quality assurance_G;
- metodi di validazione;
- processi di configurazione.

3.3 Documentazione

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 e verifica di tali documenti.

Verranno definite norme riguardanti:

- templates_G da utilizzare per i documenti, l'uniformità di linguaggio, le convenzioni stilistiche e tipografiche;
- metodi di verifica e approvazione del materiale prodotto;
- metodi di quality assurance_G.

Per agevolare la redazione della documentazione è stato creato un template_G L^AT_EX contenente tutte le impostazioni stilistiche e grafiche citate in questo documento. Tale modello si può trovare nel repository_G in docs/template.

3.3.1 Ciclo di vita

La redazione di tutti i documenti viene assegnata dal *Responsabile_G* ai responsabili della stesura di tale documento secondo i Ruolo_G definiti nella sezione ???. Successivamente se i redattori ritengono che il documento sia completo, su conferma da parte del *Responsabile_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 *Responsabile_G* l'approvazione di tale documento. Se il documento viene approvato è giunto alla fase finale.

Un documento può trovarsi in tre stati diversi:

- **In lavorazione:** un documento entra in questa fase nel momento della sua creazione, e qui vi rimane per tutto il periodo necessario alla sua realizzazione, o per eventuali successive modifiche;
- **Da verificare:** una volta che il documento viene ultimato, esso deve essere preso in consegna dai verificatori che avranno il compito di rilevare e correggere eventuali errori e/o imprecisioni sintattici e semantici;
- **Approvato:** ogni documento, una volta ultimata la fase di verifica, deve essere approvato dal Responsabile di Progetto. L'approvazione sancisce lo stato finale del documento per la data versione.

Durante la sua vita, ogni documento può attraversare ogni fase più di una volta: nel momento in cui un documento approvato necessita di una revisione formale, esso inizia nuovamente il ciclo che, al suo termine, porterà ad una nuova versione dello stesso.

3.3.2 Nome dei file

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

NomeDocumento_vX.Y.Z.pdf

dove NomeDocumento rappresenta il nome del documento e vX.Y.Z la versione come spiegata nella sezione ???. 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_G 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:

Nome Documento vX.Y.Z

3.3.3 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 *Responsabile_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 *Responsabile_G*.

Conclusa la verifica, il *Responsabile_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.

3.3.4 Stato

I documenti sono provvisti di uno stato. Tale stato può essere informale o formale.

- **Informale:** tutti i documenti saranno da ritenersi informali fino all'approvazione del *Responsabile_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.
- **Formale:** i documenti approvati dal *Responsabile_G* si riterranno formali e pronti per essere distribuiti. Solo i documenti formali potranno essere distribuiti alla loro lista di distribuzione. Ogniqualvolta un documento formale venga modificato o rivisitato, la nuova versione è da considerarsi non formale fino ad approvazione del *Responsabile_G* quindi sarà trattata come un documento informale.

3.3.5 Uso

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

- **Interno:** i documenti definiti ad uso interno non devono essere distribuiti all'esterno del gruppo di lavoro.
- **Esterno:** i documenti definiti ad uso esterno devono essere distribuiti secondo i criteri della lista di distribuzione solamente se sono stati approvati dal *Responsabile_G* e se sono in stato formale.

3.3.6 Struttura dei documenti

I documenti devono rispettare una struttura prefissata. Per agevolare la redazione della documentazione è stato creato un template_G 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`.

3.3.6.1 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 o formale)
- uso del documento (interno o esterno);
- 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;

3.3.6.2 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.

3.3.6.3 Indice

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

3.3.6.4 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.

3.3.6.5 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.

3.3.6.6 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.

3.3.6.7 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.

3.3.7 Norme tipografiche

3.3.7.1 Stile del testo

- **Corsivo:** deve essere usato solo per indicare riferimenti agli standard, citazioni, Ruolo_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

3.3.7.2 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 e il nome dell'elemento deve essere in grassetto seguito da due punti
- **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_G:* $\backslash\text{AM}$
 - *Analista_G:* $\backslash\text{AN}$
 - *Progettista_G:* $\backslash\text{PG}$
 - *Programmatore_G:* $\backslash\text{PR}$
 - *Verificatore_G:* $\backslash\text{VR}$
 - *Responsabile_G:* $\backslash\text{RE}$
- **Glossario:** ogni termine che possa causare ambiguità deve essere marcato da una lettera G scritta a pedice e riportata nel glossario. Sono state definite delle macro in \LaTeX per facilitare la composizione dei documenti:
 - $\backslash\text{mgls}$ per parole del glossario normali;
 - $\backslash\text{mGls}$ per parole del glossario che devono iniziare con la maiuscola;
 - $\backslash\text{mglspl}$ per parole del glossario plurali;
 - $\backslash\text{mGlspl}$ per parole del glossario plurali che devono iniziare con la maiuscola.
- **Note:** le note a piè di pagina possono contenere link o URL e devono essere indicate con il comando \LaTeX : $\backslash\text{footnote}$
- **\LaTeX :** i riferimenti a \LaTeX devono essere fatti utilizzando il comando \LaTeX : $\backslash\text{LaTeX}$

- **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*_G: \RR
 - *Revisione di progettazione*_G: \RP
 - *Revisione di qualifica*_G: \RQ
 - *Revisione di accettazione*_G: \RA
- **Documenti:** i riferimenti ai documenti devono essere fatti utilizzando la specifica come definita nella sezione ???. 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
 - *Definizione di prodotto*: \DP
 - *Specifica tecnica*: \ST
 - *Manuale utente*: \MU
- **Attività:** sono state definite macro L^AT_EX per i documenti in modo da facilitare la redazione dei documenti per i riferimenti alle attività del *Piano di progetto*:
 - *Analisi*: \FA
 - *Progettazione*: \FP
 - *Codifica*: \FC
 - *Validazione*: \FV

3.3.7.3 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_8601¹ 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_80000²

¹<http://www.iso.org/iso/iso8601>

²http://www.iso.org/iso/catalogue/catalogue_tc/catalogue_tc_browse.htm?commid=46202

- **Cifre:** le cifre devono seguire il seguente formato: 10 000.00, dove il separatore delle migliaia è rappresentato attraverso uno spazio e il separatore per i decimali è rappresentato attraverso uno punto.

3.3.7.4 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.

3.3.7.5 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).

3.3.8 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 alla fine di questo documento. Ogni occorrenza di vocaboli presenti nel glossario è marcata da una G maiuscola in pedice (e.g. pedice_G) ed se cliccata porta direttamente alla definizione. La generazione del glossario è automatizzata. E per facilitare la stesura dei termini viene fatta su un file a parte:

doc/glossario/glossario.tex

3.3.9 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*;

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

- i termini *può*, *opzionalmente* e sinonimi stretti sono da intendersi con lo stesso significato di *MAY*.

3.3.10 Strumenti

Per la stesura dei documenti si è scelto di utilizzare il linguaggio di markup \LaTeX . Il motivo principale che ha portato a questa scelta è la facilità di separazione tra contenuto e formattazione: con \LaTeX è possibile definire l'aspetto delle pagine in un file `templateG` condiviso da tutti i documenti. Altre soluzioni come Microsoft Office, LibreOffice o Google Docs non avrebbero consentito questa separazione, duplicando il lavoro di formattazione del testo e non garantendo un risultato uniforme. Il grande numero di pacchetti esistenti consente di implementare funzionalità comuni in maniera semplice. L'estensibilità di \LaTeX può essere sfruttata per creare funzioni e variabili globali che rendono la scrittura del contenuto più corretta da un punto di vista semantico. Un esempio è dato dal comando `Ruolo` che identifica ogni ruolo all'interno del progetto. Per la scrittura di documenti \LaTeX , l'editor consigliato è TeXstudio.

3.4 Configurazione

3.4.1 Versionamento

Come sistema di versionamento_G è stato scelto di usare `gitG`. I motivi principali che ne hanno determinato la scelta sono:

- è gratuito e non proprietario;
- è un repository_G distribuito con la possibilità di `commitG` e `revertG` locali;
- è cross-platform_G;
- possiede un'abbondante ma semplice documentazione;
- è già stato usato da alcuni elementi del gruppo.

3.4.2 Repository

Viene creata una repository_G su Github_G all'indirizzo

<https://github.com/lordnikolai/starware.swe15>

Per una guida dettagliata all'uso della repository_G si veda la sezione ??.

Commit Il `commitG` va fatto solo in uno stato consistente dei documenti modificati, dove per consistente si intende che le sezioni modificate siano complete, senza bisogno di altre modifiche note, o che un task sia stato completato. Ad ogni `commitG` deve essere associato un breve titolo che indichi quale azione è stata svolta (e.g. Aggiunta sezione repository_G nelle Norme di progetto, oppure Verifica e correzione delle Norme di progetto). Facoltativamente, ma è consigliato farlo, può essere inserito nel `commitG` un messaggio che contenga una descrizione dettagliata del lavoro svolto, riportando i motivi di eventuali scelte attuate.

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

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

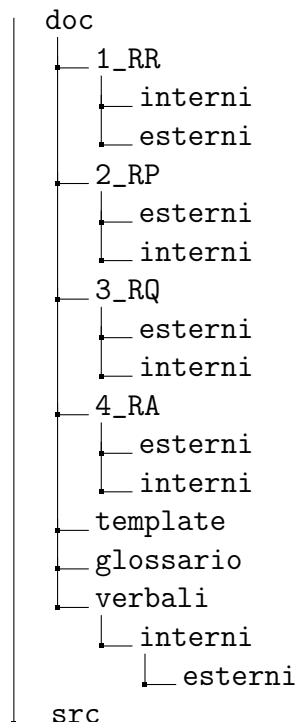
Branches vengono definiti due branches_G all'interno della repository_G :

- **master:** dove verrà tenuta la versione stabile del software e della documentazione ovvero solo i documenti verificati e approvati e solo il software con funzionalità complete che abbiano passato i test di verifica. Solo il *Responsabile_G* può fare il push_G su questo branch_G
- **naughty:** dove verranno tenuti sorgenti e documenti ancora incompleti che sono in fase di sviluppo. I documenti LATEX incompleti dovranno compilare correttamente ed avere già uno scheletro organizzativo con eventualmente sezioni vuote. Quando un documento viene verificato e poi approvato può essere messo anche nel branch_G master. Il software potrà avere funzionalità mancanti ma tutto ciò che è presente in questo branch_G deve compilare e potenzialmente non avere errori. Quando una funzionalità è completa e passa i test di verifica può esserne richiesto l'aggiornamento anche nel branch_G master

Per approfondimenti e una guida all'uso di git_G si veda <https://git-scm.com/docs/>.

3.4.2.1 Struttura albero delle directory

La repository_G del gruppo è organizzata secondo il seguente albero delle directory



La cartella `doc` contiene tutti i documenti, la cartella `src` 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_G*, *Revisione di progettazione_G*, *Revisione di qualifica_G* e *Revisione di accettazione_G*. All'interno di ognuna di queste 4 cartelle ci sono altre 2 cartelle per dividere i documenti interni (*Norme di progetto* e *Studio di fattibilità*) da quelli esterni (*Analisi dei requisiti*, *Piano di progetto* e *Piano di qualifica*).

3.4.3 Materiale informale

Per la condivisione informale di file e per il lavoro collaborativo su documenti di supporto, si usa la piattaforma di condivisione file online Google Drive_G. Trattandosi di strumenti informali, non si definiscono procedure rigorose d'uso e se ne lascia la descrizione alla sezione ??.

Google Drive In questa piattaforma di condivisione file verranno salvati i documenti che:

- non necessitano di controllo di versione;
- hanno bisogno di grande interattività tra i componenti del gruppo;
- possono essere acceduti tramite l'uso di un semplice browser.

Questo strumento dovrebbe permettere a 2 o più componenti del gruppo di interagire lavorando sugli stessi documenti contemporaneamente. L'unico vincolo posto all'utilizzo di questo strumento è di mettere a conoscenza, anche in maniera informale, tutti i membri del gruppo del materiale aggiunto.

3.5 Verifica

La verifica di processi, documenti e prodotti deve essere eseguita durante l'intero sviluppo del progetto didattico rispettando le metriche definite nel *Piano di qualifica*. 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.

3.5.1 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_G* 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 il dizionario per lo spelling_G gratuito OpenOffice vengono evidenziati e corretti gli errori di spelling_G e 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* sezione ???. 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 del glossario:** La redazione del glossario è automatizzata. Il *Verificatore_G* dovrà quindi solo controllare che tutte le voci del glossario abbiano una definizione e che tutte le parole che sono definite nel glossario siano marcate con una G a pedice.
- **Calcolo dell'indice Gulpease_G:** Su ogni documento redatto il *Verificatore_G* deve calcolare l'indice di leggibilità. Per fare ciò segnala l'utilizzo dell'applicazione web <http://labs.translated.net/leggibilita-testo/> TODO: mettere script. Nel caso in cui il testo risultasse difficile, sarà necessario eseguire un walkthrough_G del documento alla ricerca delle frasi troppo lunghe o complesse
- **Miglioramento del processo di verifica:** Quando i *Verificatori_G* eseguono walkthrough_G di un documento, dovranno riportare gli errori più frequentemente trovati nella lista di controllo_G. Grazie a tale pratica sarà possibile eseguire inspection_G su tali errori nelle verifiche future
- **Segnalazione degli errori riscontrati:** Attenendosi a quanto riportato nella tabella ?? il *Verificatore_G* deve generare dei tickets_G secondo quanto descritto nella sezione ??

3.5.2 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_G dove vengono archiviati gli errori più spesso rilevati. Man mano che il *Verificatore_G* rende più esaustiva la lista di controllo_G, si ricorre sempre meno all'utilizzo della tecnica di walkthrough_G a favore della tecnica di inspection_G;

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

3.5.3 Analisi dinamica

L'analisi dinamica si applica solamente al prodotto software e viene svolta durante l'esecuzione del codice mediante l'uso di test, per questo motivo sarà approfondita più avanti.

3.5.4 Strumenti

ProjectLibre_G viene utilizzato come strumento di controllo in quanto permette di calcolare automaticamente le metriche di Schedule variance_G e Cost variance_G. Per far ciò il *Responsabile_G* deve modificare i progetti creati nel modo seguente:

1. inserire le colonne SV e CV;
2. modificare la durata delle sotto-attività in base a quanto è stato consuntivato;
3. modificare le risorse assegnate alle sotto-attività in base a quanto è stato consuntivato;
4. aggiornare il progetto alla data corrente.

3.6 Validazione

3.6.1 Documenti

La validazione dei documenti non è altro che l'approvazione a parte del *Responsabile_G*. Il responsabile legge il documento che è appena uscito da una fase di verifica, viene letto e se viene ritenuto completo per contenuti e maturità viene validato e quindi approvato. Un documento approvato non ha bisogno di segnalazione in nessuno strumento. Esso viene semplicemente messo nel branch_G master della repository_G.

3.6.2 Requisiti

La validazione dei requisiti va tracciata con tracy_G. Quando si crea un requisito, come spiegato nella sezione ??, va inserita una descrizione che identifica il metodo di validazione. Durante la validazione i responsabili dovranno assicurarsi di applicare tale metodo per la validazione e se ritengono che il requisito è soddisfatto imposteranno il flag Apported a vero sul requisito tramite la seguente procedura:

1. eseguire l'accesso a tracy_G;
2. selezionare Requirements nella sezione Analisi Requisiti;
3. selezionare Update del requisito;
4. selezionare la checkbox Apported;

- 5.
6. premere il pulsante Save per confermare.

3.7 Risoluzione dei problemi

3.7.1 Classificazione dei problemi

Vengono classificate le categorie delle problematiche più comuni (si veda tabella ??) 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;
- software
- 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 Teamwork_G per la risoluzione del problema riscontrato (dettagli nella sezione ??);
- **Aggiunta alla lista di controllo_G:** il *Verificatore_G* scorre la lista di controllo_G 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).

Ambito	Problema	Urgenza	Azione
Processi	Ritardo di 3 giorni	Alta	Ticket
	Errore di progettazione	Alta	Ticket
	Errore in requisiti e use case	Alta	Ticket
Software	Malfunzionamento di uno strumento di lavoro	Alta	Ticket
Documentazione	Errore di compilazione	Alta	Ticket
	Errore concettuale	Alta	Ticket
	Errore in tabelle o immagini	Media	Ticket
	Gulpease _G fuori fase	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

3.7.2 Gestione dei problemi

Nel caso sorgesse un problema esso deve essere gestito tramite ticket_G. Il gruppo ha deciso di utilizzare di adottare il sistema di ticketing_G di Teamwork_G. La scelta di tale software è descritta nella sezione ??.

Ogni volta che un membro del gruppo durante riscontra un problema che, come specificato nella tabella ??, deve essere risolto tramite ticket_G apre un ticket_G di segnalazione problema.

3.7.2.1 Ticket di segnalazione problema

Il un ticket_G di segnalazione problema deve avere la seguente struttura:

- **Titolo:** Rappresenta in maniera sintetica e significativa problema
- **Assegnatari:** I ticket_G di segnalazione problema devono sempre essere assegnati al *Responsabile_G*
- **Commento:** Nel caso fossero necessarie delucidazioni al task_G bisogna chiarirle in maniera concisa sotto forma di commento
- **Priorità:** Deve essere indicata la priorità di esecuzione del ticket_G facendo riferimento a alla tabella ??

- **Etichetta/e:** Deve essere impostata l'etichetta *Problem* ed eventualmente altre etichettas_G più appropriate per aiutare visivamente i membri del gruppo a distinguere di che tipologia di ticket_G si tratta. Le etichettas_G utilizzate sono descritte nella sezione ??

3.7.2.2 Ticket di risoluzione problema

Quando il *Responsabile_G* prende visione di un ticket_G di segnalazione problema proveniente all'immediata assegnazione di un ticket_G di risoluzione problema che ha la seguente struttura:

- **Titolo:** Rappresenta in maniera sintetica e significativa problema
- **Assegnatari:** I ticket_G di risoluzione problema devono essere assegnati al o ai responsabili per il problema riscontrato
- **Data di inizio:** Impostata alla data corrente
- **Data di completamento:** Impostata in base alla priorità di esecuzione del ticket_G facendo riferimento alla tabella ??
- **Commento:** Nel caso fossero necessarie delucidazioni al task_G bisogna chiarirle in maniera concisa sotto forma di commento
- **Priorità:** Deve essere indicata la priorità di esecuzione del ticket_G facendo riferimento alla tabella ??
- **Etichetta/e:** Deve essere impostata l'etichetta *Need Fix* ed eventualmente altre etichettas_G più appropriate per aiutare visivamente i membri del gruppo a distinguere di che tipologia di ticket_G si tratta. Le etichettas_G utilizzate sono descritte nella sezione ??

Il o i responsabili dovranno risolvere il problema secondo la data di completamento assegnata.

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 amministrazione

I processi di amministrazione contengono le attività generiche e i task, che possono essere impiegati da un qualsiasi gruppo che deve gestire i suoi rispettivi processi. Il *Responsabile_G* responsabile per la gestione del prodotto, la gestione del progetto, la gestione dei task_G inerenti ai processi come quelli di acquisizione, fornitura, sviluppo, manutenzione o i processi di supporto. Verranno definite norme riguardanti:

- l'identificazione delle modalità di comunicazione interne ed esterne al gruppo;
- catalogazione dei ruoli e i loro compiti specifici.

4.1.1 Comunicazione

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 *Responsabile_G* ha accesso all'indirizzo ed è quindi l'unico a poter comunicare con il committente del progetto. È compito del *Responsabile_G* informare i membri del gruppo delle discussioni avvenute.

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_G Google Drive_G. Inoltre è permesso l'uso di Skype_G per effettuare conferenze di gruppo o anche tra singoli membri di esso.

4.1.2 Riunioni

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

Convocazione riunione Il *Responsabile_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 *Responsabile_G* che a cui spetta il compito di vagliarla. È inoltre possibile e auspicabile che possano essere necessarie riunioni tra specifici membri senza richiedere la presenza di persone non direttamente coinvolte, che verranno comunque informate delle decisioni prese tramite verbale. Il *Responsabile_G* deve convocare la riunione attraverso la creazione di un ticket_G su Teamwork_G avente le caratteristiche descritte nella sezione ?? di questo documento. Ogni componente del gruppo deve confermare la propria presenza al *Responsabile_G* il

prima possibile. In caso di mancata risposta o di immotivata assenza il *Responsabile_G* ha il dovere di contattare anche telefonicamente colui che non ha fornito una risposta in tempo utile. Il *Responsabile_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 ticket_G su teamwork_G.

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, ruolo che deve essere scelto di volta in volta a rotazione tra i presenti, deve inoltre controllare che venga seguito l'ordine del giorno in modo da non tralasciare alcun punto. Ogni verbale redatto dovrà essere inserito nell'apposita cartella **verbali** nei repository_G e dovrà essere nominato **verbale_data**, dove la data deve seguire il formato indicato nella sezione ?? di questo documento.

Riunione interna Il verbale di riunione interna è un documento interno informale che consente di tracciare gli argomenti discussi durante la riunione.

Riunione esterna In caso di riunione con il committente o 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_G L^AT_EX 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.

4.1.3 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 Responsabile

Il *Responsabile_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*, 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* e collabora alla stesura del *Piano di qualifica*, 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*, dove spiega e norma l'utilizzo degli strumenti, redige la sezione del *Piano di qualifica* 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*, l'*Analisi dei requisiti* e parte del *Piano di qualifica*. Partecipa alla redazione del *Piano di qualifica* 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*.

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* e attenendosi a quanto descritto nelle *Norme di progetto*. 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* che illustra l'esito e la completezza delle verifiche e delle prove effettuate.

4.1.3.7 Procedure per la rotazione dei ruoli

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 *Responsabile_G* e qualsiasi altro ruolo o tra *Verificatore_G* e qualsiasi altro ruolo.

4.1.4 Pianificazione

Per pianificare le attività legate allo sviluppo del progetto e la gestione delle risorse si è scelto di utilizzare ProjectLibre_G. ProjectLibre_G è un programma open-source_G per il project management_G. Tale software è stato scelto in quanto possiede le seguenti caratteristiche:

- essendo basato su Java_G, è compatibile con i sistemi operativi Linux_G, Mac OS_G e Windows_G e permette quindi ad ogni componente di installarlo senza vincoli di piattaforma;
- open-source_G;
- genera diagrammi Gantt_G;
- calcola i parametri Schedule variance_G (SV) e Cost variance_G (CV);
- salvataggio su file XML_G: essendo un file testuale è possibile effettuare il merge_G di più file in caso di conflitti sul repository_G.

Per l'uso di ProjectLibre_G si veda la sezione ??.

4.1.4.1 ProjectLibre_G

Il *Responsabile_G* deve creare un progetto per ogni fase indicata nella sezione Pianificazione del *Piano di progetto* e procedere nel modo seguente:

1. creare un calendario lavorativo per il progetto;
2. inserire le attività da svolgere e le corrispondenti sotto-attività;
3. inserire le dipendenze temporali tra le attività;
4. inserire i periodi di slack_G dove previsto;
5. inserire la milestone_G per indicare il termine previsto delle attività;
6. creare le risorse;
7. assegnare ad ogni attività le risorse necessarie;
8. salvare la baseline_G.

4.1.4.2 Teamwork

Il *Responsabile_G*, una volta individuate e stimate le attività_G e i task_G 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 Teamwork_G per i seguenti motivi:

- fornisce un sistema flessibile per la gestione dei tickets_G
- permette di creare e assegnare task_G in maniera facile ed intuitiva;
- permette di monitorare e modificare semplicemente le scadenze dei task_G creati;
- offre la possibilità di impostare delle milestones_G per tenere facilmente sotto controllo lo svolgimento delle attività_G.

Sarà inoltre compito del *Responsabile_G* assegnare una deadline_G per il completamento dei tasks_G e risolvere eventuali problemi legati a ritardi nel completamento degli stessi. Le modalità di utilizzo di teamwork_G vengono descritte nella sezione ?? di questo documento.

4.1.5 Ticketing

Per monitorare lo sviluppo dei documenti e del codice si è scelto di adottare il sistema di ticketing_G di Teamwork_G. La scelta di tale software è descritta nella sezione ??.

4.1.5.1 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 devono sempre essere assegnati ad uno o più membri del gruppo
- **Data di inizio:** Ogni tasks_G deve sempre avere una data, decisa attraverso i diagrammi di Gantt_G entro il quale essere iniziato
- **Data di completamento:** Ogni tasks_G deve sempre avere una data, decisa attraverso i diagrammi di Gantt_G entro il quale essere completato
- **Commento:** Nel caso fossero necessarie delucidazioni al task_G bisogna chiarirle in maniera concisa sotto forma di commento
- **Priorità:** Deve essere indicata la priorità di esecuzione del ticket_G in caso si trattasse di un problema facendo riferimento alla tabella ??
- **Etichetta/e:** Deve essere fornito di una o più appropriate etichettas_G per aiutare visivamente i membri del gruppo a distinguere di che tipologia di ticket_G si tratta. Le etichettas_G utilizzate sono descritte nella sezione ??

4.1.5.2 Etichette

Le etichettas_G servono per identificare meglio il ticket_G creato, quelle in uso dal gruppo sono:

- **Task:** riservata ai tickets_G di task_G
- **Code:** riservata ai tickets_G riguardanti il codice
- **Document:** riservata ai tickets_G riguardanti la documentazione
- **Check:** riservata ai tickets_G di riguardanti la verifica
- **Problem:** riservata ai tickets_G per la segnalazione problema
- **Need Fix:** riservata ai tickets_G per la risoluzione problema
- **Missing Feature:** riservata ai tickets_G per la segnalazione la mancanza di una funzionalità del software o della documentazione

4.1.5.3 Ticket di Task

Ogni ticket_G rappresenta un task_G o un'insieme di tasks_G (attivita_G) e deve rispettare i seguenti passi:

1. Creazione di un task_G;
2. Il *Responsabile_G* assegna il task_G ad uno o più membri del gruppo, nel caso l'assegnatari/o non ne abbia ancora preso visione o non abbia ancora deciso se accettare il task_G, lo stato del ticket_G sarà **Aperto**;
3. L'assegnatario può decidere se accettare o meno il task_G:
 - Accettazione: lo stato del task_G sarà **Accettato**;
 - Rifiuto: L'assegnatario dovrà motivare il rifiuto del task_G e, nel caso le motivazione fossero ritenute pertinenti dal *Responsabile_G*, verrà ripetuto il punto 2.
4. Se il task_G fosse dipendente dal completamento di un altro task_G in stato di esecuzione, allora lo stato del task_G sarà **In Attesa**;
5. Si procede con l'esecuzione del task_G, il cui stato passerà a **In Esecuzione**;
6. Una volta terminata l'esecuzione del task_G, lo stato sarà **Completato**;
7. Ogni task_G completato dovrà essere vagliato attraverso le procedure di verifica.

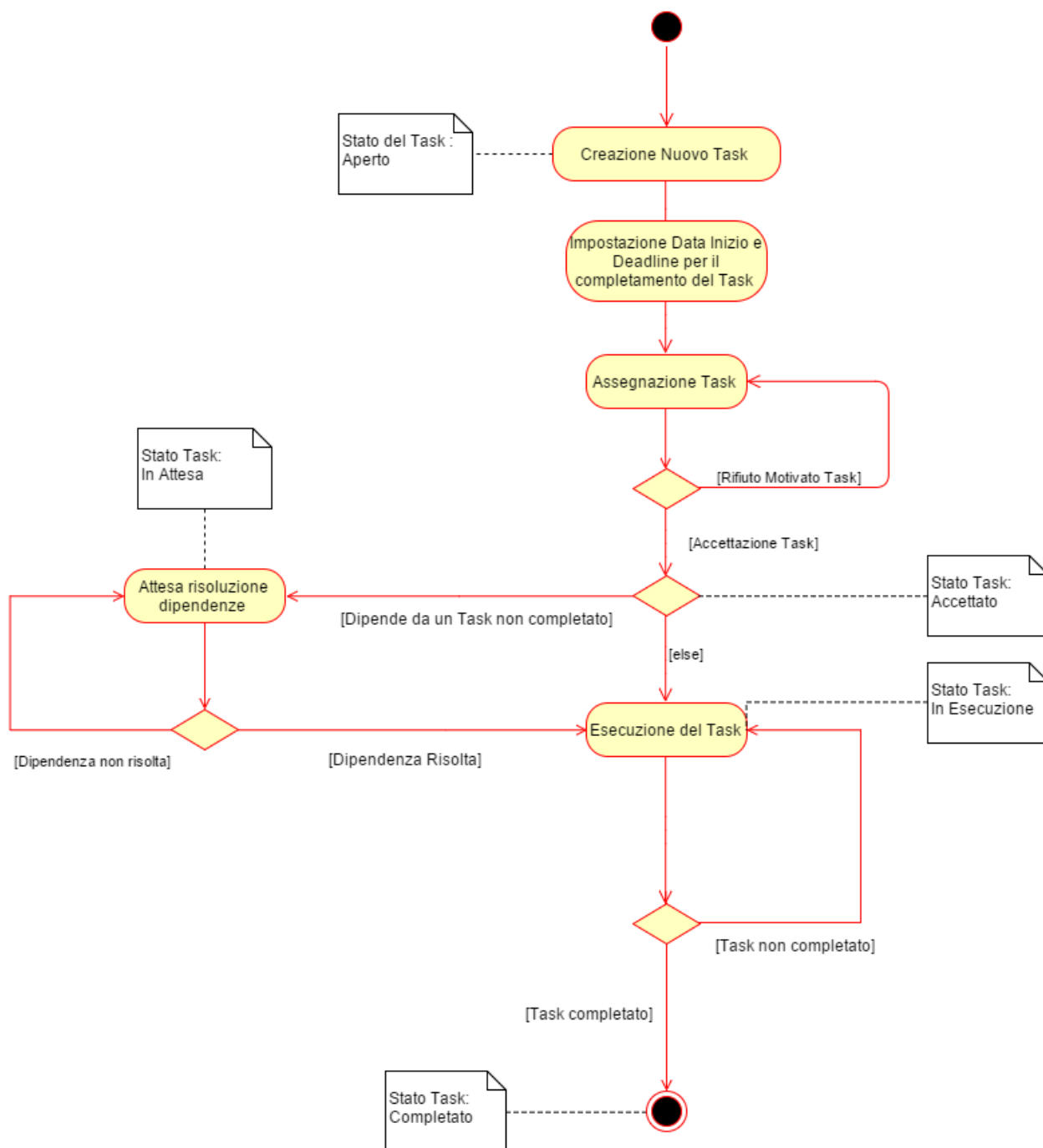


Figura 1: Diagramma del ciclo di vita dei ticket.

4.1.5.4 Ticket di Riunione

Un ticket_G di riunione è creato attraverso l'impostazione di un evento_G su Teamwork_G e deve avere le seguenti caratteristiche:

- **Titolo:** Riunione n. X, dove X indica il numero crescente di riunioni effettuate;
- **Data:** la data del ticket_G dovrà essere impostata alla data prevista per la riunione;

- **Ordine del Giorno:** elenco ordinato delle varie voci da esaminare sotto forma di commento al ticket_G;
- Deve essere assegnato a tutti i membri del gruppo.

A Lista di controllo

Durante l'applicazione del walkthrough_G ai documenti, sono state riportate le tipologie di errori più frequenti. La lista di controllo risultante è la seguente:

Norme stilistiche

- elenco puntato: non inizia con la lettera maiuscola;
- elenco puntato: non termina con il punto e virgola oppure con il punto se è l'ultimo elemento;
- elenco numerato: non termina con il punto e virgola oppure con il punto se è l'ultimo elemento;
- nome proprio di persona: non rispetta la norma Cognome Nome;
- nome ruolo di progetto: non viene utilizzata la macro predisposta;
- nome documento: non viene utilizzata la macro predisposta;
- parole Proponente e Committente: non vengono scritte con la maiuscola iniziale.

Italiano

- virgola tra soggetto e verbo: rende di difficile comprensione la frase;
- carattere È: non viene scritto correttamente utilizzando il comando apposito;
- periodi: frasi troppo lunghe rendono i concetti di difficile comprensione;
- doppie negazioni: evitare l'utilizzo di doppie negazioni perché complicano la comprensione della frase;
- punto e virgola: evitare l'uso del punto e virgola quando è necessario usare il punto;

LaTeX

- se esiste una macro per definire il termine usare sempre la macro;
- lettere accentate nelle variabili: non viene utilizzato il comando apposito;
- carattere di spaziatura: non deve essere utilizzato all'interno dei tag;
- macro \LaTeX : non viene scritta usando l'apposito comando `\LaTeX`.

UML

- Il sistema non deve mai essere un attore;
- controllo ortografico: deve essere effettuato in modo dettagliato a causa dell'impossibilità di automatizzare i controlli sui diagrammi;
- direzione delle frecce non corrette;
- consistenza della nomenclatura tra i diagrammi e le descrizioni testuali nei documenti.

La seguente lista di controllo vuole riassumere invece gli errori più frequenti rilevati durante il walkthrough_G del tracciamento requisiti effettuato mediante il software tracy_G:

Tracciamento requisiti La seguente lista vuole riassumere invece gli errori più frequenti rilevati durante il walkthrough_G del tracciamento requisiti effettuato mediante il software tracy_G:

- ad ogni caso d'uso deve corrispondere almeno un requisito;
- ad ogni requisito deve corrispondere almeno una fonte;
- la fonte "Capitolato" non deve comparire nei requisiti interni;
- deve esserci copertura totale del capitolato nei requisiti;
- devono essere impostate le corrette relazioni di parentela tra requisiti in tracy_G;
- devono essere impostate le corrette relazioni di parentela tra casi d'uso in tracy_G;
- in tracy_G deve essere indicato almeno un attore in ogni caso d'uso;
- controllare in tracy_G che le fonti dei requisiti siano le fonti corrette;
- i codici dei casi d'uso nei diagrammi e in tracy_G devono corrispondere.

