

29 dicembre 2015

StarWare

Quizzipedia: software per la gestione di questionari



Norme di Progetto

Informazioni sul documento

Nome Documento	Norme di Progetto
Versione	1.0.7
Stato	<i>In lavorazione</i>
Uso	<i>Interno</i>
Data Creazione	30 novembre 2015
Data Ultima Modifica	29 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
1.0.7	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

1	Introduzione	5
1.1	Scopo	5
1.2	Descrizione	5
2	Collaborazione	6
2.1	Comunicazione	6
2.1.1	Esterna	6
2.1.2	Interna	6
2.2	Riunioni	6
2.2.1	Frequenza	6
2.2.2	Convocazione riunione	6
2.2.3	Verbale	7
2.3	Condivisione materiale	7
2.3.1	Repository	7
2.3.2	Materiale informale	7
3	Documentazione	8
3.1	Ciclo di vita	8
3.2	Nome dei file	8
3.3	Versione	9
3.4	Stato	9
3.5	Uso	10
3.6	Struttura dei documenti	10
3.6.1	Prima pagina	10
3.6.2	Diario delle modifiche	11
3.6.3	Indice	11
3.6.4	Elenco tabelle e figure	11
3.6.5	Introduzione	11
3.6.6	Sezioni	11
3.6.7	Formattazione delle pagine	11
3.7	Norme tipografiche	12
3.7.1	Stile del testo	12
3.7.2	Composizione	12
3.7.3	Formati	13
3.7.4	Tabelle	14
3.7.5	Immagini e grafici	14
3.8	Glossario	14
3.9	Linguaggio	14
4	Ruoli	15
4.1	Project Manager	15
4.2	Amministratore	15
4.3	Analista	16
4.4	Progettista	16
4.5	Programmatore	16
4.6	Verificatore	17

4.7	Procedure per la rotazione dei ruoli	17
5	Procedure a supporto dei processi	18
5.1	Studio di fattibilità	18
5.2	Analisi dei requisiti	18
5.2.1	Scopo	18
5.2.2	Descrizione	18
5.2.3	Casi d'uso	18
5.2.4	Classificazione dei Requisiti	19
5.3	Progettazione	19
5.3.1	Scopo	19
5.3.2	Descrizione	19
5.3.3	UML	19
5.4	Verifica	20
5.5	Codifica	20
6	Ambiente di lavoro	21
6.1	Condivisione materiale	21
6.1.1	Versionamento	21
6.1.2	Repository	21
6.1.2.1	Struttura albero delle directory	22
6.1.3	Repository e versionamento	23
6.1.4	Materiale informale	24
6.2	Pianificazione	24
6.2.1	ProjectLibre	25
6.2.2	Assegnazione Task	25
6.3	Documenti	26
6.3.1	L ^A T _E X _E TeXstudio	26
6.3.2	ArgoUML	26
6.3.3	LibreOffice Calc	26
6.4	Verifica	26
6.5	Protocolli per lo sviluppo	27
6.5.1	Ticketing	27

Elenco delle tabelle

1	Versionamento del documento	1
---	---------------------------------------	---

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.

2 Collaborazione

2.1 Comunicazione

2.1.1 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.

2.1.2 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*. Per lo scambio e la condivisione di file e appunti è invece utilizzato il servizio *cloud* Google Drive_G e Dropbox_G. Inoltre è permesso l'uso di skype_G per effettuare conferenze di gruppo o anche tra singoli membri di esso.

2.2 Riunioni

2.2.1 Frequenza

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

2.2.2 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 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*, 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_G*.

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.

2.2.3 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, 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.

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

2.3 Condivisione materiale

2.3.1 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 6.1.1.

2.3.2 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 6.1.2 .

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 \LaTeX contenente tutte le impostazioni stilistiche e grafiche citate in questo documento. Tale modello si può trovare nel repository_G in `docs/template`.

3.1 Ciclo di vita

La redazione di tutti i documenti viene assegnata dal *Project Manager*_G ai responsabili della stesura di tale documento secondo i ruoli_G definiti nella sezione TODO:. 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.

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.

3.2 Nome dei file

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. 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.

Ogniquale volta 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 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.

3.4 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 *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.

- **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. Ogniqualvolta 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.

3.5 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 *Project Manager_G* e se sono in stato formale.

3.6 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.

3.6.1 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;

3.6.2 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.

3.6.3 Indice

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

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.

3.6.5 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.

3.6.6 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*.

3.6.7 Formattazione delle pagine

L'intestazione di ogni pagina, aparte 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.

3.7 Norme tipografiche

3.7.1 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

3.7.2 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
- **Membri del gruppo:** i nomi dei membri del gruppo si devono indicare con nome e poi cognome. Sono state definite macro in L^AT_EX con tutti i nomi del gruppo per facilitare la composizione dei documenti:
 - **Alessio Vitella:** \AVI
 - **Andrea Venier:** \AVE
 - **Nicola De Cao:** \NDC
 - **Igor Baylyak:** \IB
 - **Walter Sandon:** \WS
 - **Thomas Pigarelli:** \TP
 - **Anna Bonaldo:** \AB
- **Ruoli:** i nomi dei ruoli devono essere riportati in corsivo e con la prima lettera maiuscola. Sono state definite macro in L^AT_EX con tutti i ruoli definiti per facilitare la composizione dei documenti:
 - **Amministratore:** \AM

- **Analista:** \AN
- **Progettista:** \PG
- **Programmatore:** \PR
- **Verificatore:** \VR
- **Project Manager:** \PM
- **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:
 - \mgls per parole del glossario normali;
 - \mGls per parole del glossario che devono iniziare con la maiuscola;
 - \mglspl per parole del glossario plurali;
 - \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 : \footnote
- **\LaTeX :** i riferimenti a \LaTeX devono essere fatti utilizzando il comando \LaTeX : \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 ???. Sono state definite macro \LaTeX 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

3.7.3 Formati

- **Nome files:** per i nomi dei file deve essere utilizzato il comando \LaTeX : \fileName
- **Percorsi:** per i percorsi dei file deve essere utilizzato il comando \LaTeX : \filePath
- **URLs e Links:** per gli indirizzi web completi deve essere utilizzato il comando \LaTeX : \url, invece per i link deve essere utilizzato il comando \LaTeX : \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

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.

3.7.5 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 immagini e i grafici in formato vettoriale è preferito il formato Portable Network Graphics (PNG_G).

3.8 Glossario

TODO:

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

¹<http://www.iso.org/iso/iso8601>

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

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

4 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 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.

4.2 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.

4.3 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.

4.4 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*.

4.5 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.

4.6 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.

4.7 Procedure per la rotazione dei ruoli

TODO: .

5 Procedure a supporto dei processi

5.1 Studio di fattibilità

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

- Aspetti positivi
- Fattori di rischio

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

5.2 Analisi dei requisiti

5.2.1 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.

5.2.2 Descrizione

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

5.2.3 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

5.2.4 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, 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*

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.

5.3 Progettazione

5.3.1 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

5.3.2 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

5.3.3 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

5.4 Verifica

TODO: Contestualmente al PdQ

5.5 Codifica

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

6 Ambiente di lavoro

6.1 Condivisione materiale

6.1.1 Versionamento

Come sistema di versionamento_G 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_G 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

6.1.2 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`.

Commit Il comando per fare un commit è `git commit`. Il commit 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 commit deve essere associato un breve titolo che indichi quale azione è stata svolta e il numero del issue espresso come collegamento ipertestuale (e.g. Aggiunta sezione Repository nelle Norme di progetto #123, Verifica e correzione delle Norme di progetto #567). Facoltativamente, ma è consigliato farlo, può essere inserito nel commit un messaggio che contenga una descrizione dettagliata del lavoro svolto, riportando i motivi di eventuali scelte attuate.

Pull È preferibile fare un `git 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 un `git push` della repository sulla directory online. Controllare che il commit rispetti le regole del paragrafo precedente.

Branch vengono definiti 3 *branches* all'interno della repository_G:

- **stable:** 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

- **daily:** dove verranno tenuti sorgenti e documenti ancora incompleti. 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 stable*. Il software potrà avere funzionalità mancanti ma tutto ciò che è presente in questo *branch* deve compilare e potenzialmente non avere errori. Quando una funzionalità è completa e passa i test di verifica può essere messa anche nel *branch stable*
- **naughty:** dove verranno tenuti sorgenti e documenti ancora in fase di sviluppo. Quando un documento ha uno scheletro o ne viene completata una sezione può essere messo anche nel *branch stable*

dove mettiamo tutti i documenti e il software funzionante della ultima milestone (quindi con funzionalità complete al 100) e poi un branch daily dove si pushano e un branch naughty dove è permesso pushare anche cose non funzionanti così se qualcuno ha bisogno di una mano su una parte di codice che non riesce a far funzionare può pushare sul branch naughty dove ciò è permesso e farsi aiutare da qualcun'altro. Poi quando un insieme di funzionalità diventano complete sulla daily vengono verificate e se passano i test messe sulla stable.

Per approfondimenti e una guida a git si veda <https://git-scm.com/docs/>

6.1.2.1 Struttura albero delle directory

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

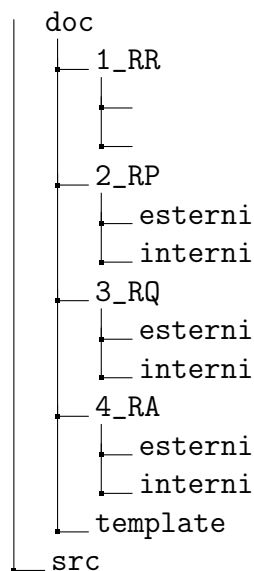
- doc
 - 1_RR
 - interni
 - esterni
 - 2_RP
 - interni
 - esterni
 - 3_RQ
 - interni
 - esterni
 - 4_RA
 - interni
 - esterni
 - template
 - glossario
 - verbali
 - interni
 - esterni
- src

6.1.3 Repository e versionamento

Sono state presi in considerazione diversi software di versionamento (Git , SVN, MerGcurial) prima di decidere di usare Git . I motivi principali della scelta sono:

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

La repository è organizzata secondo il seguente albero delle cartelle:



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_G* e *Studio di fattibilità_G*) da quelli esterni (*Analisi dei requisiti_G*, *Piano di progetto_G* e *Piano di qualifica_G*).

Si descrive ora la procedura di corretto utilizzo del repository;

- Clonare il repository_G : è possibile clonare il repository_G remoto in locale attraverso il comando `git clone sw-eng@ntp.myopenproject.it:nome-repository.git`;
- Sincronizzare il repository_G : per scaricare in locale le modifiche bisogna eseguire il comando `git pull` nella cartella che contiene il repository_G locale;
- Salvare una modifica in locale: con il comando `git add NomeFile` si aggiunge un file al commit_G corrente e usando il comando `git commit -m "Messaggio di commit"` si salvano le modifiche effettuate ai file;

- Inviare le modifiche al repositoryG remoto: tramite il comando `git push` si inviano le modifiche al repositoryG remoto;
- Gestione dei conflitti: nel caso si verifichino conflitti sarà necessario usare il comando `git mergetool` e risolvere il conflitto tramite un tool di mergeG .

Per approfondimenti e una guida a git si veda [TODO:](#) .

6.1.4 Materiale informale

Si è inoltre scelto di utilizzare degli strumenti online che permettono di condividere file in modo semplice e veloce e che consentono di organizzare gli appuntamenti personali dei singoli componenti del gruppo.

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. Google Drive viene utilizzato come strumento di supporto allo sviluppo della documentazione e del software presente su Git. 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.

6.2 Pianificazione

Per pianificare le attività legate allo sviluppo del progetto e la gestione delle risorse si è scelto di utilizzare ProjectLibreG. ProjectLibreG è un programma *open-sourceG* per il *project managementG*. È basato su JavaG ed è quindi eseguibile su ogni sistema operativo. Tale software è stato scelto in quanto possiede le seguenti caratteristiche:

- essendo basato su JavaG, è compatibile con i sistemi operativi LinuxG, Mac OSG e WindowsG. Permette quindi ad ogni componente di installarlo senza vincoli di piattaforma;
- open-sourceG;
- genera diagrammi ganttG;
- genera automaticamente diagrammi *Program Evaluation and Review TechniquesG* (PERTG a partire dal ganttG);
- genera automaticamente la *Work Breakdown StructureG* (WBS), a partire dal GanttGt con allocazione di risorse;
- calcola i parametri *Schedule VarianceG* (SV) e *Budget VarianceG* (BV);
- salvataggio su file XMLG: essendo un file testuale è possibile effettuare il mergeG di più file in caso di conflitti sul repositoryG.

6.2.1 ProjectLibre

Il *Project Manager_G* deve creare un progetto per ogni fase indicata nella sezione Pianificazione del *Piano di progetto_G* 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*.

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

1. inserire le colonne SV , BCWS, ACWP;
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.

6.2.2 Assegnazione Task

Il *Project Manager_G*, una volta individuati e stimati 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 Asana_G per i seguenti motivi:

- 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 suddividere i *task_G* fornendo un livello di *subtasking_G*.

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

6.3 Documenti

6.3.1 \LaTeX TeXstudio

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 template 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.

6.3.2 ArgoUML

TODO: .

6.3.3 LibreOffice Calc

Per l'elaborazione dei dati si utilizza il software Calc del pacchetto LibreOffice in quanto tale prodotto è open-source. Calc viene utilizzato per elaborare i dati prodotti con ProjectLibre 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 come somma delle singole righe della colonna SV di ProjectLibre;
- calcolo della metrica BVG come somma delle varie differenze tra la colonna BCWS e ACWP di ProjectLibre;
- tabelle per il confronto tra preventivo e consuntivo;
- istogrammi per il confronto tra ore preventivate e ore realmente impiegate da ciascuna risorsa.

6.4 Verifica

Per la verifica dei documenti si utilizzano:

- TeXstudio: Per la scrittura di documenti è consigliato utilizzare l'ambiente grafico TeXstudio. Tale strumento integra i dizionari di OpenOffice.org e segnala i potenziali errori ortografici presenti nel testo durante la stesura del testo stesso.
- Script: i componenti del gruppo hanno scritto vari script per il calcolo dell'indice Gulpease e la marcatura dei termini presenti nel glossario del documento.

6.5 Protocolli per lo sviluppo

6.5.1 Ticketing

