



SIRIUS

SEQUENZIATORE

Norme di progetto

Versione 1.0.0

Ingegneria Del Software AA 2013-2014

Informazioni documento

Titolo documento:	Norme di progetto
Data creazione:	2014-02-05
Versione attuale:	1.0.0
Utilizzo:	Interno
Nome file:	<i>NormeDiProgetto_v1.0.0.pdf</i>
Redazione:	Santangelo Davide
Approvazione:	Quaglio Davide
Distribuito da:	Sirius

Sommario

Il presente documento descrive le norme adottate dal gruppo *Sirius* durante la realizzazione del prodotto software *Sequenziatore*.

Diario delle modifiche

Diario delle modifiche

Versione	Data	Autore	Ruolo	Descrizione
1.0.0	2014-02-09	<i>Quaglio Davide</i>	<i>Responsabile</i>	Approvazione del documento
0.1.0	2014-02-09	<i>Giachin Vanni</i>	<i>Verificatore</i>	Verifica del documento
0.0.5	2014-02-07	<i>Santangelo Davide</i>	<i>Amministratore</i>	Stesura del documento
0.0.1	2014-02-05	<i>Seresin Davide</i>	<i>Amministratore</i>	Creato lo scheletro del documento

Indice

1	Introduzione	1
1.1	Scopo del documento	1
1.2	Ref. Glossario	1
2	Comunicazioni e relazioni inter-personali	2
2.1	Comunicazioni interne	2
2.2	Riunioni interne	2
2.3	Comunicazioni Esterne	3
2.4	Incontri esterni	3
3	Processo di sviluppo dell'applicazione	4
3.1	Gestione di progetto	4
3.2	Utilizzo del ticketing	4
3.3	Creazione di una milestone	5
3.4	Procedura di creazione ticket	6
3.5	Procedura di terminazione ticket	7
3.6	Procedura per la modifica o riassegnazione ticket	7
4	Processo di pianificazione	9
4.1	Software per la pianificazione	9
4.2	Procedura di pianificazione	9
5	Processo di documentazione	10
5.1	Template	10
5.2	Classificazione documenti	10
5.2.1	Documenti formali	10
5.2.2	Documenti informali	10
5.3	Versionamento documenti	10
5.4	Struttura Documentazione	11
5.4.1	Header	11
5.4.2	Footer	11
5.4.3	Prima pagina	11
5.4.4	Seconda pagina	12
5.4.5	Terza pagina	12
5.4.6	Quarta pagina	12
5.5	Norme tipografiche	12
5.5.1	Generali	12
5.5.2	Punteggiatura	13
5.5.3	Ortografia	13

5.5.4	Stile	13
5.6	Calcolo indice di Gulpease	14
5.7	Glossario	14
6	Processo di analisi	15
6.0.1	Classificazione dei requisiti	15
6.0.2	Casi d'uso	15
6.1	Strumento per il tracciamento	16
6.1.1	Inserimento requisito e relativo tracciamento	16
6.1.2	Visualizza script	17
7	Codifica files	18
7.1	Convenzioni di codifica	18
8	Strumenti per la coordinazione	19
8.1	Gestione condivisione file	19
8.2	Repository	19
8.2.1	GitHub	19
8.2.2	Struttura repository	19
9	Strumenti per la documentazione	21
9.1	LaTeX	21
9.2	Macros	21
9.3	Scripts	21
9.4	Correttezza	21
9.4.1	Correttezza ortografica	21
9.4.2	Lista controllo errori	22
9.4.3	Errori stilistici e di punteggiatura	22
9.4.4	Errori ortografici e di sintassi	22
9.5	UML	23
10	Ruoli di progetto	24
10.1	Responsabile di progetto	24
10.2	Amministratore	24
10.3	Analista	25
10.4	Verificatore	25
10.5	Progettista	25
10.6	Programmatore	26

1 Introduzione

1.1 Scopo del documento

Questo documento è stato redatto al fine di definire e standardizzare tutte le norme che ogni componente del team *Sirius* dovrà adottare durante il periodo di svolgimento del capitolato *Sequenziatore*, commissionato dalla *Zucchetti S.P.A.*. In particolare si andranno di seguito ad elencare convenzioni e norme per:

- Le comunicazioni e le relazioni inter-personali;
- Il processo di sviluppo dell'applicazione;
- Il processo di pianificazione del progetto;
- Il processo di *Analisi*;
- La redazione e verifica della correttezza dei documenti;
- L'utilizzo degli strumenti dell'ambiente di lavoro.

Ogni membro del gruppo *Sirius* ha l'obbligo di visionare questo documento. Tutti i componenti del gruppo sono tenuti a sottostare alle norme qui descritte di modo da garantire un corretto e coerente insieme di file prodotti, assicurando conseguentemente un continuo miglioramento dell'efficienza nello sviluppo. Infine, qualora lo si ritenesse necessario, ogni membro del team potrà contattare l'*Amministratore* per discutere norme esistenti o per valutarne l'adozione di nuove.

1.2 Ref. Glossario

Al fine di rendere più leggibile e comprensibile i documenti, i termini tecnici, di dominio, gli acronimi e le parole che necessitano di essere chiarite, sono riportate nel documento *Glossario_v1.0.0.pdf*.

Tutte le prime occorrenze di vocaboli presenti nel *Glossario* devono essere seguite da una "G" maiuscola in pedice.

2 Cooperazione

2.1 Comunicazioni interne

L'apparato di comunicazioni interne ufficiali sarà gestito tramite una *mailing list*_G basata su *Google Groups*.

Il nome della *mailing list* è Sirius esattamente come il nome del gruppo(*Sirius*).

Vige l'obbligo di utilizzare la *mailing list* solamente per le comunicazioni interne ufficiali, così da evitare intasamenti superflui che graverebbero sul lavoro di verbalizzazione delle comunicazioni di rilievo, in quanto renderebbero più complessa ed inutilmente lunga l'estrazione delle informazioni utili. Tuttavia è preferibile che la comunicazione tra i vari componenti del team avvenga principalmente durante gli incontri che si terranno in un luogo fisico comune (sezione 2.2, Riunioni interne).

Al fine di facilitare anche le comunicazioni informali, sono stati adottati due strumenti di *instant messaging*_G e videoconferenza quali Skype e Google Plus ed un strumento di *web storage* per lo scambio di dati ufficiosi (sezione 8.1, Gestione condivisione file).

2.2 Riunioni interne

Le riunioni del gruppo *Sirius* avranno una frequenza almeno settimanale. Il giorno della settimana, il luogo e l'ora in cui riunirsi ufficialmente sarà deciso dal *Responsabile di Progetto* su consultazione degli altri membri del team e sarà comunicato tramite il gruppo *Google Groups* Sirius. Chiunque non abbia la possibilità di essere presente fisicamente nel luogo della riunione, dovrà possibilmente restare in contatto con il nucleo dei membri mediante videoconferenza.

Qualunque membro del team può richiedere al *Responsabile di Progetto* di indire una riunione allegandone l'argomento di discussione ed una sua breve descrizione, a seguito della comunicazione il *Responsabile di Progetto* deciderà se indire la suddetta riunione generale, cioè con obbligatoria presenza di tutti i membri del gruppo. Qualsiasi riunione surplus a quella settimanale deve essere indetta con almeno 2 giorni di anticipo, in modo da verificare la disponibilità del gruppo.

Se dovessero essere necessarie riunioni che non richiedono la presenza del gruppo nella sua totalità, ogni membro potrà presentare la richiesta di ritrovo tramite l'apposita *mailing list* (sezione 2.1, Comunicazioni interne), richiedendo la disponibilità degli specifici membri del team che riterrà necessari, questo poiché è auspicabile che alcune figure come ad esempio *Progettista* ed *Analista* collaborino tra di loro frequentemente. Le riunioni che non coinvolgono interamente il team non necessitano dell'approvazione del *Responsabile di Progetto* in modo da ridurre il suo carico di lavoro, nonostante questo le decisioni effettuate durante queste discussioni inter-membri dovranno comunque essere verbalizzate.

2.3 Comunicazioni Esterne

Per le comunicazioni esterne di ogni tipo è stato creato in indirizzo *e-mail* del team:

swesirius@gmail.com

Il *Responsabile di Progetto* rappresenta il team stesso, sarà quindi incaricato di mantenere i contatti con proponente, committente, ed eventuali altre figure non facenti parte del nucleo del *team*, tramite questo indirizzo *e-mail*, inoltre sarà sempre parte del suo compito aggiornare i membri del gruppo stesso riguardo le corrispondenze pervenute attenendosi alle istruzioni della sezione 2.1, Comunicazioni interne.

2.4 Incontri esterni

Il *Responsabile di Progetto* ha inoltre l'onere di organizzare eventuali incontri esterni (per chiarificazioni o quant'altro) con *Proponente* o *Committente/i*. Ogni membro del gruppo può richiedere un incontro esterno al *Responsabile di Progetto*, presentando una motivazione valida. Infine, il *Responsabile di Progetto* dopo aver valutato personalmente la proposta, dovrà presentarla al gruppo (con allegata la motivazione ed il nome di chi l'ha richiesta) quindi, per l'approvazione definitiva di quest'ultima, almeno due membri escluso l'artefice dovranno dare ulteriore conferma e disponibilità, in caso contrario la proposta sarà bocciata.

3 Documentazione

3.1 Template

Ogni documento dovrà essere generato includendo il *template* \LaTeX presente nella cartella Modello. Questo modello è stato creato prima dell'inizio della redazione di ogni altro documento del team *Sirius*, la sua modifica può avvenire solo presentando all'*Amministratore* una richiesta formale, allegandone la motivazione ed il tipo di modifica richiesta. Se l'*Amministratore* riterrà opportuno effettuare il cambiamento, prima di apportare la modifica dovrà avvertire l'intero team al fine di evitare disguidi.

3.2 Classificazione documenti

3.2.1 Documenti formali

Sono catalogati come formali tutti i documenti approvati dal *Responsabile di Progetto*, ovvero i documenti ritenuti pronti per essere visionati dal committente. Tali documenti, prima di raggiungere l'approvazione dovranno aver superato con successo la procedura di verifica e validazione riportata nel *Piano di Qualifica*.

3.2.2 Documenti informali

Tutti i documenti che non sono stati approvati dal *Responsabile di Progetto* sono da ritenersi informali, e di utilizzo esclusivamente interno. Tutti i documenti non versionati sono da ritenersi non ufficiali.

3.3 Versionamento documenti

Il versionamento di tutta la documentazione del gruppo *Sirius* è stato organizzato secondo le seguenti convenzioni:

- Il numero di versionamento deve essere nella forma:

$$\mathbf{X}, \mathbf{Y}, \mathbf{Z}$$

con \mathbf{X} , \mathbf{Y} , \mathbf{Z} numeri interi non negativi;

- Tutti gli elementi devono salire di una sola unità alla volta.

Di seguito vengono inoltre riportati i significati che possono assumere le variazioni della versione del documento:

- La \mathbf{X} rappresenta il numero di uscite formali del documento, ogni qual volta un documento verrà pubblicato il valore della cifra \mathbf{Y} e della cifra \mathbf{Z} verrà azzerato. Riportando quanto detto più precisamente:

1. X assumerà il valore: 1, alla **revisione dei requisiti**;
 2. X assumerà il valore: 2, alla **revisione di progettazione**;
 3. X assumerà il valore: 3, alla **revisione di qualifica**;
 4. X assumerà il valore: 4, alla **revisione di accettazione**.
- La **Y** rappresenta il numero di *push* effettuati sul *branch_G master* in *GitHub* (sezione 8.2.1, *GitHub*), ossia il numero di volte in cui sono state compiute importanti modifiche al documento. Ogni qual volta aumenterà l'indice **Y** si azzererà l'indice **Z**.
 - La **Z** rappresenta il numero di modifiche minori apportate al documento durante il suo sviluppo. Aumenta al termine di ogni sessione di lavoro sul documento.

Ogni documento formale riporterà un diario delle modifiche contenente le trasformazioni più rilevanti che ha attraversato sotto forma tabellare.

3.4 Struttura Documentazione

3.4.1 Header

Ogni pagina esclusa la prima presenta un *header* raffigurante il logo del gruppo sulla sinistra, mentre sulla destra il nome del team ed il nome del progetto.

3.4.2 Footer

Ogni pagina esclusa la prima presenta un *footer* riportante il nome del documento corredato della versione sulla sinistra, mentre sulla destra il numero della pagina. Per il numero di pagina delle prime quattro facciate saranno utilizzati i numeri romani, a seguire invece verranno utilizzati i numeri occidentali.

3.4.3 Prima pagina

La prima pagina di ogni documento conterrà:

- Il logo del *team*, riportante la scritta *Sirius*;
- Il titolo del progetto;
- Il nome del documento e la sua versione;
- Il nome del corso;
- L'anno di sviluppo del progetto;

3.4.4 Seconda pagina

La seconda pagina di ogni documento conterrà:

- Informazioni sul documento come segue:
 - Titolo del documento;
 - Data di creazione;
 - Versione attuale;
 - Utilizzo, che specifica se il documento è per utilizzo interno o esterno;
 - Nome file;
 - Redazione;
 - Revisione;
 - Approvazione;
 - Distribuito da, a cui seguirà il nome del gruppo.
- Un sommario riportante una breve descrizione;

3.4.5 Terza pagina

La terza pagina di ogni documento conterrà:

- Un diario delle modifiche apportate al documento, dall'inizio fino alla versione corrente.

3.4.6 Quarta pagina

La quarta pagina di ogni documento ne riporterà l'indice, è possibile che l'indice si estenda per più di una singola pagina.

3.5 Norme tipografiche

3.5.1 Generali

- Ogni documento deve essere in lingua italiana, altre lingue possono essere utilizzate per riferirsi a termini tecnici informatici o in situazioni che lo richiedono strettamente;
- Ogni documento deve essere grammaticalmente, sintatticamente e semanticamente corretto, cercando di essere meno verboso possibile;
- Utilizzare il più possibile elenchi puntati invece di lunghe frasi.

3.5.2 Punteggiatura

- Non si usa mai un punto alla fine di un titolo: di capitolo, di paragrafo, di sotto-paragrafo;
- Ogni elemento di un elenco puntato termina con un punto e virgola, se è l'ultimo elemento con un punto;
- Prima di ogni segno di punteggiatura non va mai messo uno spazio bianco, dopo invece lo spazio bianco va messo sempre;
- Il testo racchiuso tra parentesi non deve aprirsi o chiudersi con un carattere di spaziatura né terminare con un carattere di punteggiatura.

3.5.3 Ortografia

- Le lettere maiuscole vanno poste solo all'inizio di ogni elemento di un elenco puntato e dove lo prevede l'ortografia italiana (all'inizio di un periodo o dopo un segno di punteggiatura forte, cioè dopo il punto fermo, i puntini di sospensione, il punto esclamativo ed il punto interrogativo). È inoltre utilizzata l'iniziale maiuscola nel nome del team, del progetto, dei documenti, dei ruoli di progetto.

3.5.4 Stile

- Se si devono elencare delle istruzioni in serie o una divisione in paragrafi e sotto-paragrafi è necessario utilizzare un elenco numerato, altrimenti è preferibile un elenco puntato;
- Il primo livello di profondità degli elenchi puntati è contrassegnato da un pallino nero pieno, il secondo da un trattino, il terzo da un asterisco;
- Le date dovranno essere espresse nella forma **aaaa-mm-gg** secondo lo standard **ISO G 8601:2004**;
- Gli orari dovranno essere espressi nella forma **hh:mm** secondo lo standard **ISO G 8601:2004**;
- **URL** ed indirizzi mail dovranno essere preceduto dal comando `LATEX \url`;
- Ogni prima (e possibilmente anche successiva) occorrenza di una parola presente sul *Glossario* sarà seguita da pedice _G.
- Stile di testo:
 - Il corsivo deve essere utilizzato obbligatoriamente nelle citazioni, per il nome delle figure di rilievo (es. *committente*, *Responsabile di Progetto*) e per il

nome dei documenti (es. *Analisi dei requisiti*), mentre a discrezione del redattore per termini stranieri in modo da evidenziarli;

- il grassetto deve essere utilizzato per evidenziare (se si reputa necessario) le parole chiave ed i passaggi particolarmente rilevanti.

3.6 Calcolo indice di Gulpease

In ogni documento redatto il verificatore dovrà calcolare l'indice di Gulpease, ossia il valore di leggibilità del documento. Per raggiungere il seguente scopo è disponibile uno *script online*, reperibile al sito:

<http://www.xoomer.virgilio.it/roberto-ricci/variabilialeatorie/esperimenti/leggibilita.htm>

Questo *script* già esistente è stato verificato prima di essere adottato, in modo da scongiurare il rischio di incompatibilità tra i documenti redatti e la forma che doveva avere l'input per lo *script*. Se l'indice risultante di un documento si troverà in un range compreso tra lo 0 ed il 40, sarà necessario ricercare nel testo frasi troppo lunghe e complesse per reimpostarle.

3.7 Glossario

Durante la stesura di un documento, ogni qual volta il redattore riterrà necessario chiarire il significato di un termine utilizzato sarà tenuto ad aggiungerlo nel *glossario*. Il *glossario* sarà strutturato seguendo questo schema:

- Nel file \LaTeX ogni parola sarà contenuta nel **tag**: elemento;
- A seguire, andando a capo-riga, sarà riportata la descrizione del termine.

Il *glossario* sarà inoltre suddiviso in due sezioni:

- Termini;
- Acronimi.

Termini ed acronimi dovranno essere necessariamente elencati in ordine alfabetico, la definizione dovrà essere breve ed esplicativa, inoltre sempre la definizione non potrà iniziare con una **E accentata**.

4 Analisi

4.0.1 Classificazione dei requisiti

Ogni requisito deve essere definito tramite una descrizione testuale e un codice identificativo, classificante e univoco avente la seguente forma:

$$\{\text{Tipologia}\}\{\text{Importanza}\}\{\text{Categoria}\}\{\text{Identificatore}\}$$

- Tipologia:
 - **F**: requisito funzionale;
 - **Q**: requisito di qualità;
 - **V**: requisito di vincolo;
- Importanza:
 - **OB**: requisito obbligatorio;
 - **DE**: requisito desiderabile;
 - **OP**: requisito opzionale;
- Categoria:
 - **U**: requisito funzionale riguardante la parte utente;
 - **L**: requisito funzionale riguardante la parte utente autenticato;
 - **A**: requisito funzionale riguardante la parte amministratore;
 - vuoto se il requisito è di qualità oppure di vincolo;
- Identificatore è un codice gerarchico composto da uno o più numeri separati da un punto, in cui l'ultimo numero è un identificatore incrementale intero.
La rimanente parte di codice viene utilizzata quando il requisito da definire è sotto-requisito di un altro, e identifica il requisito gerarchicamente superiore.

4.0.2 Casi d'uso

Per ciascun caso d'uso deve essere fornito un codice identificativo, una descrizione testuale e un diagramma UML.

Il codice identificativo deve rispettare la seguente forma:

$$\text{UC}\{\text{Tipologia}\}\{\text{Identificatore}\}$$

Tipologia può essere U, L o A, che stanno rispettivamente per utente non autenticato, utente autenticato e amministratore.

Identificatore è un codice gerarchico composto da uno o più numeri separati da un punto, in cui l'ultimo numero è un identificatore incrementale intero. La rimanente parte di codice viene utilizzata quando il caso d'uso da definire è una specifica o estensione di un altro, e identifica il caso d'uso gerarchicamente superiore.

La descrizione deve contenere i seguenti dettagli:

- Descrizione del caso d'uso;
- Attori coinvolti;
- Precondizione;
- Scenario principale dello svolgersi degli eventi;
- Senari alternativi;
- Post-condizione.

Il diagramma deve rispettare le regole della notazione UML 2.x_G.

5 Progettazione

Questa sezione descrive le norme cui i progettisti sono tenuti ad attenersi durante la progettazione dell'applicazione.

5.1 Tecnica di progettazione

5.2 Diagrammi

Per quanto concerne i diagrammi sarà [U+FFFD]ottato il linguaggio di modellazione UML 2.0, tramite questo linguaggio si andranno a definire:

- **Diagrammi dei package:** ossia elementi di raggruppamento di classi. Tali elementi dovranno figurare durante la progettazione generale e dovranno essere identificati univocamente al fine di stabilire come i suddetti interagiscono tra di loro.
- **Diagrammi delle classi:** che consentono di descrivere tipi di entità [U+FFFD]e loro caratteristiche e come queste entità [U+FFFD]teragiscono tra di loro.
- **Diagrammi di sequenza:** da vedere
- **Diagrammi di attività [U+FFFD]** atti a mostrare i flussi di attività [U+FFFD]e i vari tipi di utenti potranno compiere all'interno dell'applicazione.

5.3 Design pattern

Per ogni design pattern utilizzato sarà [U+FFFD]ecessario definire:

- Una **descrizione generale** che riporta la struttura del design pattern scelto;
- Una **motivazione** che descriva i vantaggi che ne comporta il suo uso;
- Il **contesto applicativo** che associa al design pattern il contesto ove [U+FFFD]ato adottato.

5.4 Architettura generale

6 Procedure e protocolli

6.1 Protocollo di sviluppo del progetto

6.1.1 Utilizzo del ticketing

Le figure che avranno accesso al sistema di *ticketing* sono le seguenti:

- Il *Responsabile di Progetto* assegnerà i *ticket* di massima importanza cioè quelli correlati allo sviluppo delle attività necessarie all' avanzamento del progetto;
- Il *Verificatore* potrà assegnare *ticket* allo scopo di segnalare errori di grave entità rilevati durante l'attività di verifica.

Di conseguenza i *ticket* sono suddivisi in due macro-categorie:

- *Ticket* di pianificazione, i quali rappresentano le attività che devono essere svolte per procedere con l'avanzamento del progetto, sono suddivisi in 4 sotto-categorie:
 - *Documento*: che rappresenta una *task* inerente alla redazione di un documento;
 - *Codice*: che rappresenta una *task* inerente alla stesura di codice;
 - *Verifica*: che rappresenta una *task* inerente all'attività di verifica di un'attività;
 - *Generali*: che rappresenta *tasks* i cui scopi sono svariati ed in genere non ad alta priorità, come ad esempio la ricerca di un determinato *software*.

lo svolgimento dell'insieme di tutti i *ticket* di una *task-list* non porterà alla conclusione della *task-list* stessa, questo poiché è prevista la possibilità di aggiungere durante l'avanzamento del progetto ulteriori *task*, fino a quando il *responsabile di progetto* non ne dichiarerà la conclusione;

- *Ticket* di verifica, contenenti gli errori identificati dai *verificatori* a seguito dell'analisi del lavoro svolto da qualche membro del *team*.

Ogni membro del *team* sarà tenuto ad utilizzare la barra di avanzamento di stato del *ticket* fornita dall'interfaccia di *TeamWorkPM*, evitando così superflue norme aggiuntive atte a determinare lo stato del *ticket*.

6.1.2 Creazione di una milestone

Il *Responsabile di Progetto* dovrà creare una *milestone*, essa indica la data della revisione a cui il gruppo *Sirius* intende presentarsi, è possibile visualizzare lo stato di avanzamento che tiene conto del numero di *ticket* completati rispetto al numero di *ticket* complessivi. Per la creazione di una nuova *milestone* il *Responsabile di Progetto* dovrà seguire i seguenti passi:

Add a new milestone

MARCH
3
Monday

+ Add Description...

When is it due?
03/03/2014

Who's responsible?
Davide Santangelo (Me)
Multiple People

Privacy
Everybody on project

Reminder?
☐ (An email reminder will not be sent to the person responsible)

Notify now?
☐ (The person responsible will not be notified)

Add this milestone or Cancel

Figura 1: Creazione di una milestone.

1. Aprire il progetto dall'interfaccia *web* di *TeamWorkPM*;
2. Posizionarsi sull'opzione: *Milestones* ed accedervi;
3. Cliccare sull'opzione: *Add a new milestone*.

Completati questi passaggi apparirà il seguente *form_G* (fig 1) che dovrà essere compilato per concludere la creazione della una *milestone*.

6.1.3 Procedura di creazione ticket

Il *Responsabile di Progetto* dovrà attenersi alla seguente procedura per la creazione di una nuova *task-list*, ovvero la concretizzazione di un macro-attività e delle sue relative *task* (ticket). Si ricorda che *TeamWorkPM* prevede la possibilità di indicare interdipendenze tra *task-list*.

1. Dall'interfaccia *web* accedere al progetto *Sequenziatore*, e selezionare dal *menù* principale il comando: *Task*;
2. Procedere, se necessario, con la creazione di una nuova *task-list* tramite il comando: *Add task list*;

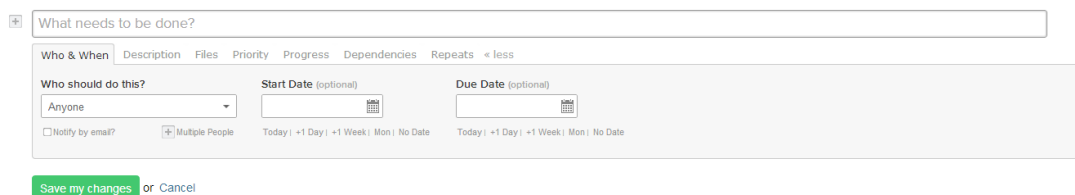


Figura 2: Creazione di un ticket.

- Una volta creata la *task-list* sarà possibile creare i *ticket* (*Task* nel contesto di *TeamWorkPM*) inerenti alla *task-list* scelta.

La struttura di un *ticket* è visualizzabile nella figura 2 di questa sezione.

Nella task è necessario specificare **obbligatoriamente**:

- Il **Titolo** del *ticket*, dovrà contenere tra parentesi quadre la categoria (per i *ticket* di pianificazione anche la sotto-categoria) di *ticket* di cui si tratta;
- Il **Destinatario** del *ticket*, cioè colui a cui è stato assegnato;
- Le **Date** di inizio e scadenza del *ticket*;
- Le **Dipendenze** del *ticket*, che specificano l' eventuale necessità di attendere la terminazione di un insieme di *task* prima di poter svolgere quel determinato compito;
- Una **Descrizione** la quale dovrà essere breve e concisa, ma spiegare efficacemente il lavoro assegnato;
- La **Priorità** del *ticket* suddivisa in tre categorie: bassa, media, alta.

Compilati i seguenti campi, il *ticket* sarà creato ed inviato regolamentarmene.

6.1.4 Procedura di terminazione ticket

Se un *ticket* sarà completato è necessario applicare questa procedura di accertamento:

1. Il membro del *team* a cui è stato assegnato il *ticket* dovrà spuntare la casella di terminazione su *TeamWorkPM*;
2. Durante il controllo giornaliero il *Responsabile di Progetto* controllerà quanto necessario a determinare che il lavoro sia stato effettivamente svolto;
3. Se il lavoro è stato effettivamente svolto, sarà avviato un *ticket* di *Pianificazione* a scopo di verificare il lavoro;

4. Nel caso di irregolare svolgimento del *ticket* o problemi di grave entità, il *Responsabile di Progetto* dovrà applicare la procedura di modifica o riassegnazione del *ticket* presente nel prossimo paragrafo;
5. Nel caso di esito positivo (cioè con regolare svolgimento) il *ticket* sarà concluso ed archiviato, mentre al contempo, qualora fosse necessario, saranno avviati dei *ticket* di *Verifica* per la correzione degli errori non gravi rilevati durante la *Verifica*;

6.1.5 Procedura per la modifica o riassegnazione ticket

Durante il suo ciclo di vita un *ticket* per varie ragioni può andare in contro a modifiche, è necessario quindi normare la seguente procedura:

1. Aprire il progetto dall'interfaccia *web* di *TeamWorkPM*;
2. Selezionare il *ticket* di interesse;
3. Selezionare il comando: *Edit Task*;
4. Aggiungere una descrizione riguardo la modifica effettuata;
5. Avvertire l'interessato che è stata effettuata una modifica inserendo: (MOD) sul titolo del *ticket*, e qualora fosse necessario reimpostandone la sua priorità.

6.2 Protocollo di pianificazione

Il *responsabile di progetto* per ogni attività indicata nel documento *Piano di Progetto* dovrà creare un nuovo progetto seguendo la procedura qui descritta:

1. Inserire una milestone_G;
2. Inserire le attività da svolgere;
3. Inserire le rispettive sotto-attività;
4. Calcolare ed inserire i periodi di slack_G qualora fosse necessario;
5. Creare le risorse;
6. Assegnare le risorse create ad ogni attività;
7. Salvare la baseline_G.

Sarà decisa a discrezione del *Responsabile di Progetto* per ogni attività la possibilità di assegnare un surplus di ore, queste ore supplementari verranno scelte basandosi sulla criticità dell'attività considerata.

- Per le attività non critiche non è previsto alcun surplus di ore;

- Per le attività di media criticità il surplus di ore potrà essere del 15%;
- Per le attività di criticità massima il surplus di ore potrà essere del 30%.

7 Codifica files

Si prevede di ampliare questa sezione nel momento in cui inizierà la vera e propria attività di codifica. Ci si propone ora di fornire le più importanti linee guida riguardo le convenzioni che saranno adottate dal *team*.

7.1 Convenzioni di codifica

Al fine di produrre codice ordinato e leggibile, in modo da semplificare il più possibile l'attività di manutenzione, per quanto concerne la programmazione *JavaG*, si adotteranno le norme imposte dalla *Java code conventions*, reperibili all'indirizzo:

www.oracle.com/technetwork/java/codeconvtoc-136057.html

Variazioni e modifiche a queste convenzioni possono essere richieste all'*Amministratore*, allegandone la motivazione. Se l'*Amministratore* riterrà opportune le variazioni presentate, sarà tenuto a notificarlo al team seguendo le convenzioni imposte dalle *Norme di Progetto*.

8 Strumenti di lavoro

8.1 Strum. documentazione

8.1.1 LaTeX

Per la stesura dei documenti il team *Sirius* ha deciso di adottare il linguaggio di markup \LaTeX la scelta è stata effettuata prevalentemente per le seguenti ragioni:

- Facilità di separazione tra contenuto e formattazione;
- Possibilità di definire macro ed incorporare scripts;
- Software open source;
- Grande quantità di pacchetti disponibili, possibilità quindi di implementare semplicemente le funzionalità comuni.

Gli altri software valutati (Open office, Microsoft Office, Google Docs) non erano in grado di fornire il più delle sopracitate funzionalità, di conseguenza sono stati scartati. Inoltre come editor è consigliato ma non obbligato l'uso di TeXstudio.

8.1.2 Macros

Al fine di velocizzare il lavoro di stesura documenti, il team *Sirius* ha deciso di creare delle apposite macro, qui vengono riportate le principali assieme ad una breve spiegazione delle loro funzionalità:

- \backslash gruppo riporta il nome del team *Sirius*;
- \backslash progetto riporta il nome del progetto: *Sequenziatore*;
- \backslash lastversion+sigla-del-documento riporta il nome del documento che appare in sigla (NDP, AR, etc..) aggiornato alla versione più recente.

8.1.3 Scripts

Al fine di implementare una funzionalità quale l'inserimento automatico dei pedici in tutte le parole dei documenti formali che comparivano anche nel glossario, è stato creato uno script apposito in Python_G. Tale script può essere eseguito solamente con una versione di Python non superiore alla 2.7.6. Per il corretto funzionamento dello script il glossario è stato organizzato tramite tag \LaTeX elementoparola del glossario, la definizione è riportata a capo-riga rispetto alla suddetta parola.

8.1.4 Correttezza

8.1.4.1 Correttezza ortografica Per evitare di compiere errori di tipo ortografico devono essere adottate due precauzioni:

- Verifica delle parole durante la stesura stessa del documento tramite lo spell checker_G di TeXstudio;
- Verifica finale tramite lo spell checker Aspell.

Lo spell checker di TeXstudio è una sua feature_G molto utile che sfrutta dizionari Open Office per sottolineare eventuali parole scorrette, dizionari sufficientemente completi che assicurano quindi un grado piuttosto elevato di correttezza già durante la stesura del testo.

Al fine poi di assicurarsi il massimo grado possibile di correttezza viene effettuata una verifica ulteriore tramite il software open source GNU Aspell (www.aspell.net).

8.1.4.2 Lista controllo errori Il team ha stilato una lista di controllo al fine di riassumere gli errori più ricorrenti in ogni documento, i suddetti saranno catalogati e descritti nella seguente sezione.

8.1.4.3 Errori stilistici e di punteggiatura I principali errori rilevati sono i seguenti:

- Le figure di rilievo non vengono scritte in corsivo;
- Negli elenchi puntati la prima parola non compare con la prima lettera maiuscola;
- Negli elenchi puntati alcuni elementi centrali non terminano con un punto e virgola ma con un punto fermo;
- Alcune date vengono erroneamente scritte senza seguire lo standard **ISO G 8601:2004**;
- La parola LaTeX compare senza l'utilizzo del comando `\LaTeX` (`\LaTeX`).

8.1.4.4 Errori ortografici e di sintassi

- La è accentata compare (erroneamente) come una e apostrofata;
- Utilizzando le seguenti macro `\gruppo` e `\progetto`, le quali scrivono testualmente e rispettivamente il nome del team ed il nome del capitolato, non compaiono separate dalla parola successiva, anche se la spaziatura è presente;
- Non viene utilizzata (erroneamente) la terza persona per la stesura dei documenti.

8.1.5 UML

Per la modellazione dei diagrammi User Case (UC_G) sono stati presi in considerazione tre editor: Dia, Microsoft Visio, Astah. Infine il team ha optato per adottare Astah come strumento definitivo in quanto si tratta di un software open source, con supporto di Unified Modeling Language (UML_G) 2.x e secondo l'analisi del team dotato di un interfaccia più responsiva ed intuitiva degli altri software.

8.2 Strum. gestione del progetto

Al fine di gestire rigorosamente lo sviluppo del progetto il team *Sirius* ha adottato l'utilizzo del software *TeamWorkPM* (www.teamwork.com), tale strumento fornisce le seguenti funzionalità:

- Creazione di $ticket_G$, $milestone_G$ e liste di attività;
- Creazione ed assegnazione di attività;
- Calendario di progetto;
- *Report* automatico giornaliero delle attività svolte ed in ritardo inviato tramite *e-mail*;
- Gestione dei ruoli;
- Generazione di grafici $Gantt_G$ a partire dalle *task-list*;
- Monitoraggio dei tempi;
- Registro dei rischi.

Sono stati valutati altri software come ad esempio *Redmine*, il quale fu ritenuto quasi altrettanto completo ed intuitivo. Tuttavia si è optato per *TeamWorkPM* data la sua estrema semplicità di utilizzo.

Il *Responsabile di Progetto* per garantire il regolare svolgimento delle attività dovrà necessariamente verificare con cadenza giornaliera se sono presenti *ticket* scaduti e non ancora completati, nel caso citato infatti sarà obbligato a richiedere informazioni (o in caso di ritardi gravi convocare l'interessato) circa la causa del ritardo.

Infine, il *Responsabile di Progetto* dovrà tenere nota che l'assegnazione di *ticket* la cui scadenza è prevista per il giorno successivo può avvenire solo se il lavoro da svolgere non supera le 2 ore lavorative.

8.3 Strum. pianificazione

Per l'attività di pianificazione del progetto nonché gestione delle risorse è stato adottato il software GanttProject, software open source basato su piattaforma Java. Qui di seguito vengono elencate le principali caratteristiche che hanno portato alla scelta di questo strumento:

- Portabilità, essendo un software basato su Java;
- Open-source_G;
- Compatibile con MicrosoftProject;
- Può generare grafici Work Breakdown Structure (WBS_G);
- Fornisce la possibilità di creare grafici di Gantt_G;
- Può generare grafici Program Evalutation and Review Tecnique (PERT_G);
- In grado di gestire e generare grafici delle risorse assegnate.

8.4 Strum. di versionamento

Di pressoché fondamentale importanza è stata la definizione di un ambiente ordinato in cui organizzare e mantenere tutti i *file* che attraversano il ciclo di vita, per questa ragione è stato scelto di avvalersi di un *repository*.

8.4.1 GitHub

Come sistema di controllo di versione è stato adottato il software *GitHub_G*, i pregi di questo strumento vengono qui di seguito riportati:

- Molto reattivo;
- Design semplice;
- *Software* gratuito.

8.4.2 Struttura repository

L'indirizzo di **root_G** del *repository* contenente tutta la documentazione è:

<https://github.com/Dquaglio/Sirius>

Ogni documento presente sarà contenuto in una sotto cartella del *master branch_G* nominata come il nome del documento stesso. All'interno della cartella potranno essere contenuti solamente *file.tex*, per la visualizzazione del relativo *pdf_G* sarà necessario scaricarli e compilarli, assicurandosi di avere l'ultima versione del modello disponibile.

Corretto macro.tex per AnalisiRequisiti		
Vgiachin authored 13 hours ago latest commit c0b080d72a		
AnalisiDeiRequisiti	Aggiornata Analisi dei Requisiti	13 hours ago
EsempioDocumento	aggiornato esempio documento	21 days ago
Glossario	ultima modifica alla grafica del Diario	22 days ago
NormeDiProgetto	Update Sviluppo.tex	4 days ago
PianoDiProgetto	ultima modifica alla grafica del Diario	22 days ago
PianoDiQualifica	bozza piano di qualifica	21 days ago
StudioDiFattibilita	terza persona nello studio fattibilità	16 days ago
Verbali	creata copia di modello dentro Verbale	17 days ago
modello	aggiornamento modello da master	13 hours ago
.gitignore	tolto .gitingore~	21 days ago

Figura 3: Struttura master branch.

modello.git: <https://github.com/Dquaglio/Sirius/tree/master/modello.git>

conterrà il *template* $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$, le *macro* e gli script aggiornati all'ultima versione disponibile.

Il *master branch* è stato quindi suddiviso seguendo questa struttura: A scopo puramente dimostrativo è stato creato l'esempio di un documento formale del gruppo *Sirius*, contenuto nella cartella: *EsempioDocumento*, questa scelta è stata fatta per illustrare la struttura generale che deve preservare qualsiasi documento (sezione 5.4, Struttura documentazione). Per sfruttare il parallelismo nello sviluppo di uno stesso documento sono stati creati appositamente dei *branch* denominati con il nome dei membri del gruppo, i documenti *baseline_G* invece saranno contenuti solamente nel *master branch*. Il *merge_G* con il ramo *master* avviene quindi solamente dopo la terminazione dell'attività di verifica di un documento.

8.5 Strum. condivisione file

Per gestire efficientemente la condivisione dei file intra-gruppo a supporto dello strumento di **Git** è stato scelto l'utilizzo di: Google Drive, un servizio *web* di *storage* e sincronizzazione *online* che dovrebbe facilitare la condivisione e fornire una base d'appoggio secondaria ed informale per alcuni file che non necessitano versionamento. L'utilizzo di *Google Drive* è limitato ai documenti che:

- Non necessitano di versionamento;
- Necessitano di essere acceduti velocemente tramite *web*;

8.6 Strum. tracciamento

Per il tracciamento è stato sviluppato un semplice programma denominato Sirius RTg. Questo programma è stato sviluppato utilizzando PHP_G , CSS_G ed HTML_G . Sirius RTg è attualmente un programma il cui sviluppo non è terminato, principalmente per permettere una aggiunta di funzionalità in caso di necessità. Sirius RTg, alla versione 1.4.0, fornisce le seguenti funzionalità:

- Inserimento requisito e relativo tracciamento;
- Visualizzazione dello script per la tabella dei requisiti e relativo tracciamento.

8.6.1 Inserimento requisito e relativo tracciamento

Questa funzionalità è fornita all'esterno attraverso un'interfaccia scritta in HTML e CSS.

L'interfaccia è costituita da uno semplice *form*, in cui è possibile inserire:

- Codice requisito;
- Descrizione del requisito;
- Categoria;
- Importanza;
- Tipo;
- Relativi casi d'uso;
- Relative fonti.

Ogni requisito deve avere obbligatoriamente definito il Tipo, l'Importanza, ed il Codice requisito. Il codice deve necessariamente identificare univocamente il requisito, altrimenti un uso ridondante di codici verrà notificato all'utente. Obbligatoria la categoria, in caso il requisiti sia della parte utente.

8.6.2 Visualizza script

Questa funzionalità serve per la stampa a video dei vari script in latex_G per i vari requisiti e il relativo tracciamento.

Visualizza script è composto dalle seguenti funzionalità:

- Visualizzazione script per Requisiti di tipo utente amministratore;
- Visualizzazione script per Requisiti di tipo utente-autenticato;
- Visualizzazione script per Requisiti di vincolo;

- Visualizzazione script per Requisiti di qualità;
- Visualizzazione script tracciamento Requisiti-uc;
- Visualizzazione script tracciamento Uc-requisiti.

Ogni script stampato su video, segue le regole definite nelle *Norme di Progetto*.

Anche se Visualizzazione script utente e Visualizzazione script utente autenticato sono due sotto-funzionalità distinte di Visualizza script, devono essere utilizzate assieme per produrre la tabella dei requisiti di tipo utente; infatti Visualizzazione script utente stampa l'intestazione della tabella e la parte dei requisiti utente, mentre Visualizzazione script utente autenticato stampa la parte dei requisiti utente autenticato e i comandi necessari per chiudere la tabella.

Tutti gli altri Script, invece possono essere usati singolarmente e a video, oltre al contenuto comparirà la relativa intestazione e chiusura della tabella.

9 Ruoli di progetto

9.1 Responsabile di progetto

Il *responsabile di progetto* è colui che ha potere decisionale sul progetto, l'entità quindi che ha l'incarico di approvare le scelte effettuate ed il lavoro svolto. Questo potere decisionale si ripercuote sulla responsabilità che detiene nel presentare il risultato del prodotto creato al proponente. Il responsabile di progetto deve inoltre occuparsi della gestione (assegnazione, modifica) dei *ticket*, ed assicurarsi che i verificatori seguano sistematicamente le regole imposte dalle *Norme di progetto*. Riassumendo, tale figura ha responsabilità sotto il profilo di:

- Approvazione dell'offerta economica;
- Approvazione dei documenti;
- Analisi e gestione delle risorse;
- Analisi e gestione dei rischi;
- Pianificazione del piano di lavoro;
- Coordinazione del team;
- Controllo del regolare svolgimento delle attività;
- Evitare conflitti di interesse tra redattori e verificatori.

Per ciò che concerne la documentazione, il responsabile di progetto è colui che deve redigere il *Piano di progetto* e collaborare nella stesura del *Piano di qualifica*.

9.2 Amministratore

La figura dell'*Amministratore* è responsabile per tutto ciò che è pertinente all'efficienza ed operatività dell'ambiente di lavoro. I suoi compiti di primaria importanza sono qui di seguito riportati:

- Gestione del versionamento del prodotto;
- Ricerca degli strumenti più adatti allo svolgimento del progetto;
- Automatizzazione delle attività che possono essere risolte tramite strumenti e procedure automatiche, in modo da snellire il carico di lavoro ad personam;
- Creare o ricercare strumenti che possano controllare/monitorare la qualità del prodotto;

- Normare le procedure standard di pianificazione del progetto, coordinazione del team, redazione dei documenti, produzione del codice.

L'amministratore deve redigere interamente le *Norme di progetto* nonché collaborare nel *Piano di qualifica*.

9.3 Analista

L'*Analista* è competente nell'attività di analisi ed astrazione dei requisiti di progetto. Di seguito vengono riportate le sue mansioni principali:

- Astrarre i requisiti dal problema in modo da creare una specifica di progetto comprensibile dal progettista, dal proponente e dal committente;
- Comprendere i requisiti meno espliciti del problema da affrontare.

I documenti: *Analisi dei Requisiti* e *Studio di Fattibilità* devono essere stilati dall'*Analista*. Nel *Piano di qualifica* dovrà illustrare il livello di qualità richiesta e le procedure da attuare per raggiungerla.

9.4 Verificatore

Il verificatore è colui che deve effettuare l'attività di verifica. Questa figura deve attenersi alle regole imposte dalle *Norme di progetto*, e tramite l'ausilio degli strumenti illustrati nel *Piano di qualifica* avrà il compito di assicurare che:

- Le attività svolte siano coerenti agli standard adottati.
- La documentazione sia conforme alle *Norme di progetto*;

9.5 Progettista

Il progettista è colui che ha la responsabilità sull'attività di progettazione. I suoi compiti possono essere riassunti come segue:

- Agire in modo che il progetto sia sviluppato tramite tecnologie al più possibile stabili e note;
- Agire in modo che il progetto sia sviluppato seguendo soluzioni (come ad esempio soluzioni progettuali) ottimizzate e note;
- Creare una soluzione progettuale adeguata, ossia comprensibile ed attuabile;
- Agire sulle scelte progettuali in modo da sviluppare un prodotto facilmente manutenibile.

Tale ruolo avrà il compito di redigere la *Specifica Tecnica*, la *Definizione di Prodotto* e la parte inerente alla metrica di verifica nel *Piano di Qualifica*.

9.6 Programmatore

Come deducibile è la figura che si occuperà dell'attività di codifica. Le principali mansioni di questo ruolo si riassumono nei seguenti punti:

- Implementare le soluzioni progettuali specificate dal *Progettista*, senza discuterle o senza prendere alcuna iniziativa personale.
- Scrivere codice manutenibile rispettando gli standard imposti per la codifica;
- Documentare tutto il codice generato, in modo chiaro e conciso;
- Implementare i test per la verifica e validazione del codice.

Il *Programmatore* si occuperà infine di scrivere il *Manuale Utente*.