



## MaaP: MongoDB as an admin Platform

---

### Norme di Progetto

<b>Versione</b>	1.2.0
<b>Data creazione</b>	2013-11-20
<b>Data ultima modifica</b>	2014-01-27
<b>Stato del Documento</b>	Formale
<b>Uso del Documento</b>	Interno
<b>Redazione</b>	Giacomo Pinato, Andrea Perin
<b>Verifica</b>	Alessandro Benetti
<b>Approvazione</b>	Andrea Perin
<b>Distribuzione</b>	Aperture Software

#### Sommario

Questo documento si propone di presentare le norme che il gruppo **Aperture Software** ha stabilito per la Realizzazione del prodotto **MaaP: MongoDB as an admin Platform**.

---

Diario delle modifiche

Versione	Data	Autore	Modifiche effettuate
2.0.1	2014-01-29	Giacomo Pinato (RE)	Sezione Progettazione, Riorganizzazione
2.0.1	2014-01-21	Mattia Sorgato (RE)	Modifica sezione Comunicazioni
1.2.0	2013-11-27	Andrea Perin (RE)	Approvazione documento
1.1.0	2013-11-26	Alessandro Benetti (VE)	Verifica documento
1.0.3	2013-11-25	Giacomo Pinato (RE)	Ampliamento documento
1.0.1	2013-11-20	Andrea Perin (RE)	Creazione documento

Tabella 1: Registro delle modifiche

## Indice

<b>1</b>	<b>Introduzione</b>	<b>5</b>
1.1	Scopo del documento . . . . .	5
1.2	Scopo del prodotto . . . . .	5
1.3	Glossario . . . . .	5
1.4	Riferimenti . . . . .	5
<b>2</b>	<b>Ruoli di progetto</b>	<b>6</b>
2.1	Responsabile di Progetto . . . . .	6
2.2	Amministratore . . . . .	6
2.3	Analista . . . . .	6
2.4	Progettista . . . . .	7
2.5	Verificatore . . . . .	7
2.6	Programmatore . . . . .	7
<b>3</b>	<b>Comunicazioni</b>	<b>8</b>
3.1	Comunicazioni interne . . . . .	8
3.1.1	Comunicazioni interne formali . . . . .	8
3.1.2	Comunicazioni interne informali . . . . .	8
3.2	Riunioni . . . . .	8
3.2.1	Richiesta . . . . .	8
3.2.2	Svolgimento . . . . .	9
3.2.3	Verbale . . . . .	9
3.3	Comunicazioni con i committenti e proponenti . . . . .	9
3.3.1	Prerequisiti . . . . .	9
3.3.2	Richiesta di colloquio . . . . .	9
3.3.3	Svolgimento del colloquio . . . . .	9
3.3.4	Verbale . . . . .	10
<b>4</b>	<b>Ambiente e strumenti di lavoro</b>	<b>11</b>
4.1	Coordinamento . . . . .	11
4.1.1	Redmine . . . . .	11
4.1.2	Google Calendar . . . . .	11
4.1.3	Jenkins . . . . .	11
4.1.4	Git . . . . .	11
4.2	Strumenti per la documentazione . . . . .	12
4.2.1	LaTeX . . . . .	12
4.2.2	Strumenti di Verifica . . . . .	12
4.3	Codifica . . . . .	12
<b>5</b>	<b>Documentazione</b>	<b>13</b>
5.1	Template . . . . .	13
5.2	Struttura del documento . . . . .	13
5.2.1	Header . . . . .	13
5.2.2	Footer . . . . .	13
5.2.3	Prima pagina . . . . .	13
5.2.4	Seconda pagina . . . . .	13
5.2.5	Terza pagina . . . . .	13

5.3	Documentazione del codice . . . . .	14
5.3.1	Header dei file . . . . .	14
5.3.2	Documentazione dei metodi . . . . .	14
5.3.3	Documentazione delle classi . . . . .	14
5.4	Norme tipografiche e stili di testo . . . . .	14
5.5	Versionamento . . . . .	16
5.5.1	Regole generali . . . . .	16
5.5.2	Versionamento del software . . . . .	16
5.5.3	Versionamento dei documenti . . . . .	16
5.6	Glossario . . . . .	18
<b>6</b>	<b>Procedure</b>	<b>19</b>
6.1	Analisi dei requisiti . . . . .	19
6.1.1	Identificazione e classificazione dei requisiti . . . . .	19
6.1.2	Casi d'uso e UML . . . . .	19
6.2	Progettazione . . . . .	19
6.2.1	Specifica Tecnica . . . . .	19
6.2.2	Definizione di Prodotto . . . . .	20
6.3	Gestione di Progetto . . . . .	20
6.3.1	Gestione delle attività . . . . .	20
6.3.2	Creare un nuovo progetto . . . . .	21
6.3.3	Ticket . . . . .	21
6.3.4	Creazione dei ticket . . . . .	21
6.3.5	Tipologie di Ticket . . . . .	21
6.4	Verifica . . . . .	23
6.4.1	Analisi dei documenti . . . . .	23
6.5	Walkthrough . . . . .	23
6.6	Inspection . . . . .	23
6.7	Calcolo indice Gulpease . . . . .	25

## 1 Introduzione

### 1.1 Scopo del documento

Questo documento è volto a definire le norme che dovranno essere osservate da tutti i componenti del team per l'intera durata del progetto. Tali norme sono volte a garantire la qualità finale del prodotto attraverso la rigida regolamentazione dei processi e delle strategie di produzione. In queste pagine vengono delineate le linee guida e le norme per tutte le attività che concorreranno allo sviluppo del *software<sub>G</sub>*, della documentazione e del progetto in generale.

### 1.2 Scopo del prodotto

Lo scopo del prodotto è produrre un framework per generare interfacce web di amministrazione dei dati di business basati sullo stack Node.js e MongoDB.

L'obiettivo è quello di semplificare il lavoro allo sviluppatore che dovrà rispondere in modo rapido e standard alle richieste degli esperti di business.

### 1.3 Glossario

Al fine di evitare ogni ambiguità nella comprensione del linguaggio utilizzato nel presente documento e, in generale, nella documentazione fornita dal gruppo Aperture Software, ogni termine tecnico, di difficile comprensione o di necessario approfondimento verrà inserito nel documento *Glossario\_v1.2.0.pdf*.

Saranno in esso definiti e descritti tutti i termini in corsivo e allo stesso tempo marcati da una lettera "G" maiuscola in pedice nella documentazione fornita.

### 1.4 Riferimenti

- **Slide dell'insegnamento Ingegneria del Software modulo A:**  
Ingegneria dei requisiti: <http://www.math.unipd.it/~tullio/IS-1/2013/>;
- **Software Engineering - Ian Sommerville - 9th Edition (2010);**
- **Standard ISO 8601 - Data elements and interchange formats:** [http://it.wikipedia.org/wiki/ISO\\_8601](http://it.wikipedia.org/wiki/ISO_8601)
- **Piano di Qualifica:** *Piano\_di\_Qualifica\_v1.2.0*;
- **Piano di Progetto:** *Piano\_di\_Progetto\_v1.2.0*.

## 2 Ruoli di progetto

Per la piena riuscita del progetto è indispensabile distinguere i vari ruoli che concorrono alla creazione del prodotto finale e le loro diverse responsabilità e competenze. Ogni ruolo avrà una specifica area di competenza, degli specifici compiti, oneri e particolari autorizzazioni. Ogni componente dovrà limitarsi ai compiti ad esso assegnati e, nel caso qualcosa esuli dal suo campo di pertinenza, lo stesso dovrà rivolgersi al collega occupante il ruolo competente. Tutti i membri, a rotazione, dovranno occupare come minimo una volta ciascuno dei ruoli descritti sottostante.

### 2.1 Responsabile di Progetto

Il Responsabile di Progetto incentra su di sé le responsabilità di scelta ed approvazione dei lavori. Ha inoltre il ruolo di rappresentare il gruppo nei contatti con l'esterno e durante la presentazione dei lavori.

Le sue competenze principali comprendono:

- Pianificazione, coordinamento e controllo delle attività;
- Gestione e controllo delle risorse;
- Approvazione delle analisi di gestione e rischio;
- Approvazione dei documenti;
- Comunicazioni con i committenti/proponenti.

Il Responsabile ha il compito di assicurarsi che le attività di Verifica vengano svolte sistematicamente seguendo le *Norme di Progetto v1.2.0*, deve al contempo accertarsi che vengano rispettati i ruoli e le competenze assegnate nel *Piano di Progetto v1.2.0* e che non vi siano conflitti di interesse tra redattori e Verificatori. Ha infine l'onere di gestire la creazione e l'assegnazione dei ticket di pianificazione e di assegnare ad un membro del gruppo il ruolo di Responsabile di quest'ultimo, nel caso riguardi una sotto-attività.

### 2.2 Amministratore

L'*Amministratore<sub>G</sub>* è il responsabile del controllo, dell'efficienza e dell'operatività dell'*ambiente<sub>G</sub>* di lavoro e degli strumenti per la condivisione e la sincronizzazione.

Inoltre deve garantire:

- L'individuazione e la gestione di strumenti per automatizzare quanto più possibile processi o attività;
- L'individuazione e la gestione di strumenti per il controllo dei processi e delle risorse;
- L'individuazione e la gestione di strumenti e strategie per il controllo della qualità;
- Gestione del *versionamento<sub>G</sub>*.

### 2.3 Analista

L'Analista è il responsabile dell'Analisi dei Requisiti di progetto. Dopo aver compreso pienamente la natura del problema e tutti i suoi domini, il suo ruolo è delineare vincoli e caratteristiche del prodotto finale, redigendo una specifica di progetto dettagliata, precisa e non ambigua, comprensibile sia dal Proponente che dal Progettista. Partecipa inoltre al Piano di Qualifica e allo Studio di Fattibilità.

## 2.4 Progettista

Il Progettista è colui che disegna una soluzione attuabile ed efficace che soddisfi i requisiti dettati dagli Analisti. I suoi compiti principali sono:

- Disegnare una soluzione fattibile e adatta ad soddisfare i requisiti delineati dagli Analisti;
- Implementare soluzioni di progettazione ottimali, ottimizzate e collaudate;
- Utilizzare tecnologie e standard che rendano facile la realizzazione e la manutenzione del prodotto.

Redige la Specifica Tecnica e la Definizione di Prodotto, oltre a partecipare alla stesura delle metriche di verifica della progettazione del Piano di Qualifica.

## 2.5 Verificatore

Il Verificatore è responsabile delle attività di Verifica. Ha il compito di assicurare che i documenti e il codice<sub>G</sub> rispettino gli standard stabiliti nel documento *Norme\_di\_Progetto.v1.2.0*. Redige la sezione del Piano di Qualifica che illustra gli esiti delle prove e delle verifiche effettuate.

## 2.6 Programmatore

Il *Programmatore<sub>G</sub>* 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 per la Realizzazione del progetto;
- Scrivere codice e la sua relativa documentazione che rispettino gli standard stabiliti per la loro scrittura;
- Implementare i test sul codice scritto, necessari per prove di Verifica e Validazione.

Redige inoltre il Manuale Utente e tutta la documentazione relativa al codice.

## 3 Comunicazioni

### 3.1 Comunicazioni interne

Le comunicazioni interne sono divise in comunicazioni formali e informali, in base alla natura della comunicazione stessa.

#### 3.1.1 Comunicazioni interne formali

Le comunicazioni interne formali appartengono all'area di organizzazione generale del gruppo. A questa categoria appartengono tutti gli avvisi di impegni collettivi del team, la segnalazione di imprevisti organizzativi o la convocazione di riunioni.

**3.1.1.1 Mailing List** E' stata creata una mailing list all'indirizzo [aperture-team@googlegroups.com](mailto:aperture-team@googlegroups.com). Ogni *email<sub>G</sub>* inviata a tale indirizzo verrà inoltrata alla email personale di ogni componente nel gruppo. La mailing list è collegata ad un gruppo su *Google<sub>G</sub>* Groups, che permette di velocizzare le comunicazioni implementando una *simil-chat<sub>G</sub>* tramite email distribuite, inoltre mantiene uno storico di qualsiasi comunicazione. Va usato quindi come strumento per discussioni e comunicazioni ufficiali.

#### 3.1.2 Comunicazioni interne informali

Le comunicazioni riguardanti dubbi o chiarimenti marginali o comunque non inerenti all'organizzazione del gruppo, vengono definiti informali. Per loro natura, queste comunicazioni devono essere veloci e non complesse, quindi abbiamo deciso di utilizzare strumenti immediati di comunicazione, quali le *chat<sub>G</sub>*. In questo caso, un membro del gruppo esprime il dubbio come messaggio istantaneo, il quale sarà visibile da ogni altro componente.

**3.1.2.1 Skype** Per facilitare le comunicazioni istantanee è stata creata su *Skype<sub>G</sub>* una chat di gruppo dove sono presenti tutti i membri del team. Tale chat deve essere usata solo per conversazioni non ufficiali o per discussioni non importanti come lo scambio di articoli, consigli o comunicazioni non rilevanti.

**3.1.2.2 Gruppo Facebook** Data la grande diffusione di questo *social network<sub>G</sub>* e l'aderenza di tutti i membri del gruppo ad esso, abbiamo deciso di creare un gruppo apposito su questa piattaforma. *Facebook<sub>G</sub>* fornisce un accesso ancora più immediato di Skype, data la sua diffusione su dispositivi mobile.

## 3.2 Riunioni

### 3.2.1 Richiesta

Ogni membro del team può richiedere che venga organizzata una riunione al Responsabile che, una volta verificata la motivazione della richiesta, accoglie o meno la stessa. Il Responsabile di Progetto ha la facoltà di indire riunioni qualora sentisse la necessità di farlo. La riunione deve essere segnalata sul calendario di gruppo con almeno due giorni di anticipo e non deve andare a sovrapporsi con impegni precedenti di altri componenti, a meno che la loro presenza non possa essere esclusa. Deve inoltre essere inviata una email di promemoria a tutti i membri del gruppo indicante giorno, ora e motivazione della riunione.



### 3.2.2 Svolgimento

Alle riunioni è gradita, ma non richiesta, la partecipazione di tutti i membri del gruppo. E' giustificata l'assenza nel caso in cui la riunione riguardi temi non inerenti al ruolo di progetto che si sta ricoprendo o nel caso di impegni validi, giustificabili e improrogabili.

### 3.2.3 Verbale

Ad ogni riunione verrà eletto un segretario che avrà il compito di annotare gli argomenti di discussione e le decisioni prese durante la stessa. Dovrà inoltre redigere un verbale che verrà ufficialmente raccolto e archiviato come documentazione interna entro tre giorni dalla data della riunione. La struttura del Verbale deve contenere i seguenti elementi:

- Data;
- Luogo;
- Ora;
- Durata;
- Partecipanti interni;
- Partecipanti esterni;
- Motivazione della riunione/incontro;
- Argomenti trattati.

## 3.3 Comunicazioni con i committenti e proponenti

### 3.3.1 Prerequisiti

Prima di richiedere un colloquio personale con i proponenti e/o committenti il team deve preparare un documento che riassume i punti che verranno discussi contenente argomenti, dubbi e domande da porre.

### 3.3.2 Richiesta di colloquio

I colloqui con i committenti di progetto possono essere richieste dal Responsabile mediante la email ufficiale del team. Tutti i membri devono essere avvisati prima di richiedere un incontro con il Committente.

**3.3.2.1 Email** La email ufficiale del gruppo è [ApertureSWE@gmail.com](mailto:ApertureSWE@gmail.com).

Tale email può essere usata soltanto dal Responsabile di Progetto e verrà utilizzata per tutte le comunicazioni che il gruppo terrà con l'ambiente esterno.

### 3.3.3 Svolgimento del colloquio

Al fine di ottimizzare l'estrapolazione delle informazioni derivanti dal colloquio, durante tutto lo svolgimento dello stesso dovrà essere registrata una traccia audio come supporto alla stesura del verbale.

**3.3.4 Verbale**

Alla fine del colloquio verrà redatto un verbale (vedi 3.3.3) che riassume gli argomenti trattati, le conclusioni, nonché le strategie che si sono delineate in concordanza col Committente/Proponente.

## 4 Ambiente e strumenti di lavoro

### 4.1 Coordinamento

#### 4.1.1 Redmine

Come piattaforma per la gestione del progetto è stato scelto Redmine. Le principali funzioni che esso fornisce sono:

- Un sistema di gestione dei ticket;
- Il grafico  $Gantt_G$  delle attività;
- Un calendario per organizzare i compiti e le attività;
- La visualizzazione del  $repository_G$  associato al progetto;
- Un sistema di gestione del tempo e delle  $milestone_G$ .

#### 4.1.2 Google Calendar

Viene messo a disposizione del team un calendario per coordinare temporalmente le attività e gli impegni di ciascuno dei componenti. Tutti i membri del team useranno il calendario messo a disposizione su Google Calendar per segnalare i giorni in cui per loro non sarà possibile lavorare al progetto o partecipare a riunioni e sessioni di lavoro di gruppo. I componenti si impegnano a segnare sul calendario quanto prima qualsiasi impegno o impedimento dovesse sorgere e a controllare periodicamente il suddetto per essere aggiornati su eventuali impegni altrui e/o di gruppo. Il calendario verrà inoltre usato dal Responsabile per la segnalazione di riunioni o sessioni di lavoro di gruppo.

#### 4.1.3 Jenkins

Jenkins è una cosa tanto bella. //ESPANDERE FFS///

#### 4.1.4 Git

Viene messo a disposizione un repository  $Git_G$  su  $GitHub_G$  per la gestione e il versionamento di codice e documenti tra i vari membri del gruppo.

Il repository pubblico è disponibile all'indirizzo:

<https://github.com/ApertureSoftware/AperturePublic.git>.

**Branch** Nel repository saranno disponibili vari  $branch_G$  per favorire lo sviluppo del codice da parte degli sviluppatori.

**Master** Il branch principale sarà chiamato master e conterrà l'ultima  $versione_G$  del software stabile rilasciata. Affinché una nuova versione possa essere caricata nel branch master, quest'ultima deve compilare senza errori o warning e deve aver superato tutti i test disegnati per verificarne la qualità.

**Secondari** Gli sviluppatori possono richiedere la creazione di un branch secondario, provando nuove strategie di sviluppo senza alterare il branch master. Il Responsabile di Progetto è Responsabile dell'approvazione della richiesta.

## 4.2 Strumenti per la documentazione

### 4.2.1 LaTeX

Per la stesura dei documenti verrà utilizzato il linguaggio di markup  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ . Esso rende possibile definire template di documenti standardizzati da poter applicare a qualsiasi contenuto, separando così formattazione da contenuto e facilitando il lavoro di stesura.  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$  inoltre dispone di qualsiasi strumento di formattazione di cui si possa avere bisogno, eliminando il bisogno di strumenti ausiliari alla stesura dei documenti.

### 4.2.2 Strumenti di Verifica

Le risorse software che si utilizzeranno durante i processi di Verifica sono:

- Correttore automatico TeXstudio: questo strumento integra i dizionari di OpenOffice.org e segnala i potenziali errori ortografici presenti nel testo;
- Aspell: strumento per la correzione tipografica dei documenti redatti in  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ ;
- Glossario-script: script, scritto dai componenti del gruppo, che marca i termini nel glossario con la simbologia sopracitata.
- Script Gulpease: script per il calcolo automatico dell'indice di Gulpease.

## 4.3 Codifica

## 5 Documentazione

### 5.1 Template

Viene fornito un *template<sub>G</sub>* in IAT<sub>E</sub>X per la Realizzazione della documentazione, sia interna che esterna, che i membri del gruppo dovranno seguire nella stesura dei documenti.

### 5.2 Struttura del documento

#### 5.2.1 Header

Ogni documento ha un header presente in ogni pagina che riporta logo e nome del gruppo sulla sinistra.

#### 5.2.2 Footer

Ogni documento ha un footer presente in ogni pagina che riporta il nome e la versione del documento a sinistra e il numero della pagina a destra.

#### 5.2.3 Prima pagina

La prima pagina di ogni documento deve riportare nel seguente ordine:

- Il logo esteso del gruppo;
- Il nome del progetto;
- Il nome del documento;
- Informazioni sul documento (versione, data creazione, data ultima modifica, stato del documento, uso del documento, i redattori del documento, i Verificatori, chi ha approvato il documento e la *distribuzione<sub>G</sub>* del documento);
- Breve sommario del documento.

#### 5.2.4 Seconda pagina

La seconda pagina deve riportare il diario delle modifiche apportate al documento dalla sua creazione fino alla versione corrente, strutturato nella seguente maniera:

Versione	Data	Autore	Modifiche effettuate
----------	------	--------	----------------------

Le modifiche dovranno essere ordinate cronologicamente dalla più alla meno recente.

#### 5.2.5 Terza pagina

La terza pagina deve riportare l'indice del documento.

## 5.3 Documentazione del codice

### 5.3.1 Header dei file

Ogni file dovrà contenere un header strutturato come segue:

- File: Nome del file;
- Module: modulo di appartenenza;
- Author: Autore (indirizzo email dell'autore);
- Created: Data di creazione;
- Version: Versione corrente;
- Description: Descrizione dettagliata del file;
- Modification History: Tabella dei cambiamenti effettuati sul file.

### 5.3.2 Documentazione dei metodi

Ogni metodo/funzione di codice dovrà essere preceduta da un commento contenente le seguenti informazioni:

- Name: Nome del *metodo<sub>G</sub>*;
- Param: lista del tipo dei parametri;
- Descr: Breve descrizione del comportamento del metodo;
- Return: Cosa ritorna la funzione.

### 5.3.3 Documentazione delle classi

Ogni *classe<sub>G</sub>* deve essere preceduta da un commento contenente le seguenti informazioni:

- Name: Nome della classe;
- Descr: Breve descrizione della classe.

## 5.4 Norme tipografiche e stili di testo

Nella stesura della documentazione si dovranno seguire le seguenti indicazioni:

- I documenti dovranno essere grammaticalmente e sintatticamente corretti e scritti in modo fluido;
- Elenco puntato termina con il punto e virgola oppure con il punto se è l'ultimo elemento;
- Un carattere di punteggiatura non deve mai seguire un carattere di spaziatura;
- Il testo racchiuso tra parentesi non deve aprirsi o chiudersi con un carattere di spaziatura e non deve terminare con un carattere di punteggiatura;
- Le lettere maiuscole vanno poste solo dopo il punto, il punto di domanda, il punto esclamativo e all'inizio di ogni elemento di un elenco puntato, oltre che dove previsto dalla lingua italiana. È inoltre utilizzata l'iniziale maiuscola nel nome del team, del progetto, dei documenti, dei ruoli di progetto, delle fasi di lavoro e nelle parole Proponente e Committente.

- Nel caso in cui ci si riferisca ad un documento, il titolo di quest'ultimo dovrà essere scritto in corsivo e si dovrà riportare la versione riferita;
- I ruoli di progetto (es. Analista) dovranno essere scritti con la lettera iniziale maiuscola;
- Gli acronimi dovranno essere scritti in lettere maiuscole;
- È preferibile usare la forma attiva a quella passiva;
- Quando possibile usare elenchi puntati invece che frasi;
- Usare termini specifici e segnare i termini del glossario in corsivo e con la G in pedice;
- Dividere i documenti in sezioni e sottosezioni titolate;
- Le date dovranno essere espresse nella forma AAAA-MM-GG secondo lo standard *ISO<sub>G</sub>* 8601:2004;
- Le attività (es. Verifica) vanno scritte con la lettera iniziale maiuscola;
- Gli elenchi puntati con un primo livello di profondità sono formati da dei pallini neri pieni, tranne quando si deve elencare una sequenza numerata di istruzioni da fare in un ordine stabilito, allora in quel caso si usa un elenco numerato;
- Gli elenchi puntati con un secondo livello di profondità hanno un trattino;
- Il numero massimo di subordinate in una frase è quattro;
- 

**Sigle e abbreviazioni** Le sigle e le abbreviazioni dovranno essere utilizzate solo in contesti in cui lo spazio è limitato come tabelle e diagrammi. Sono previste le seguenti abbreviazioni:

- AdR = Analisi dei Requisiti;
- GL = Glossario;
- NdP = Norme di Progetto;
- PdP = Piano di Progetto;
- PdQ = Piano di Qualifica;
- SdF = Studio di Fattibilità;
- ST = Specifica Tecnica;
- RA = Revisione di Accettazione;
- RP = Revisione di Progettazione;
- RQ = Revisione di Qualifica;
- RR = Revisione dei Requisiti;
- AS = Aperture Software.

## 5.5 Versionamento

Il versionamento verrà effettuato sia sul codice che sui documenti prodotti dal gruppo al fine di differenziare e rendere immediatamente riconoscibile la fase di sviluppo in cui ci si trova attualmente, mantenendo uno storico organizzato dei cambiamenti effettuati.

### 5.5.1 Regole generali

- Il numero di versionamento deve essere nella forma X.Y.Z, con X,Y e Z numeri interi non negativi. Tutti gli elementi devono salire numericamente di una unità alla volta;
- Ogni qualvolta che una versione viene rilasciata non può più essere effettuato alcun cambiamento ad essa. Ogni modifica sarà inserita nella versione successiva.

### 5.5.2 Versionamento del software

Nel versionare il software, le tre cifre di versionamento verranno modificate in base ai seguenti parametri:

- La prima cifra decimale verrà aumentata nel caso in cui vengano introdotti cambiamenti non retro compatibili al *framework<sub>G</sub>*. Possono essere inclusi cambiamenti minori. Nel caso in cui la prima cifra venga aumentata la seconda e la terza ripartono da 0;
- La seconda cifra decimale verrà aumentata nel caso in cui vengano rilasciate nuove funzionalità retro compatibili. Deve necessariamente essere aumentata se una qualsiasi funzionalità pubblica del framework viene marcata deprecata. E' possibile aumentare la seconda cifra anche in caso di rilascio di nuove funzionalità o miglioramenti sostanziali. Al cambiamento della seconda cifra la terza deve ripartire da 0;
- La terza cifra decimale verrà aumentata nel caso di correzione di errori o altri piccoli cambiamenti retro compatibili;
- La cifra X a 0 è riservata per lo sviluppo iniziale. Le versioni con X a 0 non sono da considerarsi stabili;
- La versione 1.0.0 definisce la prima versione stabile del framework.

### 5.5.3 Versionamento dei documenti

Nel versionare il software, le tre cifre di versionamento verranno modificate in base alle seguenti regole:

- X: indica il numero di uscite formali del documento, diviso come segue:
  1. Fase di Analisi, si estende fino alla Revisione dei Requisiti;
  2. Fase di Analisi in Dettaglio, si estende fino all'ingresso nella fase di Progettazione;
  3. Fase di Progettazione Architettuale, si estende fino alla Revisione di Progettazione;
  4. Fase di Progettazione di Dettaglio e Codifica, si estende fino alla Revisione di Qualifica;
  5. Fase di Verifica e Validazione, si estende fino alla Revisione di Accettazione e alla fine del progetto.
- Y: indica la fase di sviluppo del documento e varia come segue:
  0. Fase di stesura del documento, dove il documento è ancora modificabile;



1. Fase di Verifica del documento, dove il documento non è più modificabile e sta venendo controllato dal Verificatore;
2. Documento ultimato e approvato.

Sarà compito del Responsabile impostare la versione a X.1.0 nel momento in cui assegnerà il ticket di controllo ai Verificatori e impostare la versione a X.2.0 nel caso in cui il documento risulti approvato dagli stessi.

- Z: indica il numero di modifiche minori apportare al documento. Aumenta al termine di ogni sessione di lavoro sul documento. Non ha un limite massimo.

## 5.6 Glossario

Ogni qualvolta, durante la stesura di un documento, si incorra in un termine tecnico o non di uso comune, esso dovrà essere inserito nel Glossario insieme ad una sua precisa definizione. Se la definizione conterrà a sua volta termini di questo tipo, anche essi a loro volta dovranno essere inseriti nel Glossario. Le definizioni dei termini devono essere quanto più semplici e chiare possibili, e devono cercare di evitare ricorsioni o termini da glossario. I termini nel Glossario devono necessariamente essere elencati in ordine alfabetico. Ogni definizione del termine del Glossario deve contenere una spiegazione concisa; è preferibile che la definizione inizi con un sostantivo, in ogni caso non deve mai iniziare con l'espressione *è un*. All'interno della definizione del termine, l'occorrenza del termine stesso non dovrà contenere la notazione che lo identifica come termine da Glossario, ovvero scritto in corsivo e con la lettera G come pedice.

## 6 Procedure

### 6.1 Analisi dei requisiti

La stesura del documento denominato Analisi dei Requisiti è compito degli Analisti. In questo documento verranno trovati, analizzati e catalogati tutti i requisiti, impliciti e non, imposti dal progetto.

#### 6.1.1 Identificazione e classificazione dei requisiti

È compito degli Analisti stilare una lista dei requisiti emersi dal capitolato e da eventuali riunioni con il Proponente. E' inoltre loro compito identificare tutti quei requisiti impliciti che per loro natura non vengono specificati ma devono essere individuati e studiati. Tutti questi requisiti dovranno essere classificati per tipo e importanza, utilizzando la seguente codifica:

**R[importanza][tipo][codice]**

dove:

- Importanza può assumere i seguenti valori:
  - O: *Requisito<sub>O</sub>* obbligatorio;
  - D: Requisito desiderabile;
  - F: Requisito facoltativo.
- Tipo può assumere i seguenti valori:
  - F: Funzionale;
  - Q: Di qualità;
  - P: Prestazionale;
  - V: Vincolo.
- Codice è il codice che identifica univocamente ogni requisito. La sua regolamentazione è descritta nell'*Analisi\_dei\_Requisiti\_v1.2.0*.

#### 6.1.2 Casi d'uso e UML

Successivamente all'individuazione ed al tracciamento dei requisiti si procede all'analisi dei casi d'uso, denominati nelle sezioni seguenti anche come use case o con l'acronimo UC. Per realizzare i diagrammi UML dei casi d'uso è stato scelto Astah. Il software era stato consigliato dal Professor Cardin e soddisfa tutte le necessità del team. Tutti i diagrammi devono rispettare la specifica UML 2.0.

## 6.2 Progettazione

### 6.2.1 Specifica Tecnica

I Progettisti devono descrivere l'architettura d'alto livello del sistema e dei singoli componenti nella Specifica Tecnica in modo chiaro, formale e non ambiguo.

**6.2.1.1 Design Pattern** I Progettisti dovranno fornire una descrizione dei design pattern utilizzati per la progettazione dell'architettura. In essi si deve includere una descrizione ed un diagramma che mostri il funzionamento e la struttura.

**6.2.1.2 Tracciamento componenti** Ogni requisito deve essere tracciato al componente che lo soddisfa, ed ogni componente deve essere tracciato al requisito che ha imposto la sua realizzazione.

**6.2.1.3 Test di integrazione** I Progettisti dovranno definire in dei test e delle classi atte a verificare il lavoro svolto.

## 6.2.2 Definizione di Prodotto

I Progettisti devono descrivere l'architettura di basso livello del sistema e dei singoli componenti nella Definizione di Prodotto in modo chiaro, formale e non ambiguo, integrando quanto scritto nella Specifica Tecnica.

**6.2.2.1 Diagrammi UML** Devono essere realizzati i seguenti diagrammi UML

- Diagrammi delle attività;
- Diagrammi di sequenza;
- Diagrammi delle classi.

**6.2.2.2 Definizione delle classi** Ogni classe progettata deve essere riportata all'interno della Definizione di Prodotto secondo lo schema definito nei formalismi di specifica.

**6.2.2.3 Tracciamento delle classi** Ogni requisito deve essere tracciato alla classe che lo soddisfa, ed ogni classe deve essere tracciata al requisito che ha imposto la sua realizzazione.

**6.2.2.4 Test di Unità** I Progettisti dovranno definire i test d'unità necessari per verificare che i componenti del sistema funzionino nella modo previsto.

## 6.3 Gestione di Progetto

### 6.3.1 Gestione delle attività

**6.3.1.1 Pianificazione** All'inizio di ciascuna delle fasi delineate nel *Piano di Progetto v2.2.0* il Responsabile di Progetto deve realizzare un diagramma di Gantt su Redmine per tracciare e gestire le attività che andranno a cominciare. Deve inoltre assegnare ogni attività ad un componente o gruppo di componenti.

**6.3.1.2 Controllo dell'avanzamento** Dopo ogni sessione di lavoro di un componente o gruppo di componenti, gli stessi devono tracciare il loro lavoro modificando l'avanzamento dell'attività e le ore/persona spese. Dovranno inoltre modificare lo stato dell'attività in base al ruolo ricoperto e lo stato di avanzamento della stessa come descritto nel paragrafo successivo.

**6.3.1.3 Stati d'avanzamento** I possibili stati di avanzamento di un'attività sono i seguenti:

- *Nuovo*: Stato iniziale di ogni attività
- *In Elaborazione*: Il lavoro sull'attività è iniziato;
- *Chiuso*: Il lavoro è stato completato e la percentuale di completamento è al 100%;
- *Risolto*: Stato riservato ai ticket di modifica. Le correzioni effettuate hanno risolto il problema riscontrato;
- *Rifiutato*: Stato riservato ai ticket di modifica. Le correzioni effettuate non hanno risolto il problema riscontrato ed è necessario intervenire nuovamente;

### 6.3.2 Creare un nuovo progetto

La creazione del progetto è onere del Responsabile di Progetto. Il progetto è una macro-attività che verrà suddivisa in molte sotto-attività al fine di semplificare la Progettazione, la Realizzazione e la Verifica delle stesse. Ogni attività sarà a carico di un Responsabile.

### 6.3.3 Ticket

I ticket sono uno strumento per il tracciamento di attività, funzionalità e problematiche utile per sincronizzare e mantenere traccia del lavoro del team.

### 6.3.4 Creazione dei ticket

I ticket vengono creati dal:

- Responsabile di Progetto: crea i ticket di maggiore importanza per l'avanzamento tra le fasi macroscopiche di progetto;
- Responsabile di sotto-progetto: crea i ticket minori per lo sviluppo delle caratteristiche e delle funzionalità delle singole componenti del progetto;
- Verificatore: crea ticket per la segnalazione di problematiche o bug che necessitano di controllo e correzione.

### 6.3.5 Tipologie di Ticket

I ticket rappresentano operazioni e attività che devono essere portate a termine. I ticket avranno specifici stadi di avanzamento:

- *Nuovo*: il ticket è stato creato ed è in attesa di essere assegnato;
- *In Elaborazione*: il ticket è stato assegnato ed è in lavorazione;
- *Chiuso*: la lavorazione è terminata e l'attività è in attesa di verifica;
- *Risolto o Rifiutato*: un Verificatore approva o rigetta il lavoro effettuato. Nel caso di rigetto deve venire creato un nuovo ticket di modifica per risolvere il problema.

**6.3.5.1 Ticket di Realizzazione** Assegnano la realizzazione delle attività ai membri del gruppo competenti. Sono organizzati in una gerarchia con vari livelli di priorità.

**6.3.5.2 Ticket di Verifica** Rappresentano segnalazioni di errori, problematiche o imprecisioni riscontrate dai Verificatori durante le fase di Verifica. Dovranno essere assegnati ad un membro del team che in quel momento ricopre un ruolo adatto ad intervenire per la correzione e in seguito riverificati da un Verificatore.

**6.3.5.3 Ticket di modifica** Rappresentano richieste di modifica fatte da membri del team al Responsabile. Quest'ultimo poi decide se accettare la richiesta di modifica o rifiutarla. Se la richiesta viene accettata, il Responsabile convoca una riunione per la discussione della modifica. Se la modifica viene rettificata dall'assemblea, le viene assegnata una priorità e viene messa nell'elenco di modifiche in attesa di implementazione. Le priorità che possono essere assegnate sono le seguenti:

- Urgente: da applicare il prima possibile;
- Media: da applicare entro la prossima milestone;
- Bassa: l'implementazione non è obbligatoria, non c'è una scadenza definita.

## 6.4 Verifica

Vengono qui esplicitate le procedure con cui si eseguono l'analisi statica e dinamica per la Verifica dei documenti.

### 6.4.1 Analisi dei documenti

- Controllo sintattico e del periodo: utilizzando TeXstudio e GNU Aspell vengono evidenziati e corretti gli errori di grammatica più evidenti. Ciascun documento dovrà essere sottoposto ad un walkthrough da parte dei Verificatori per individuare errori di *sintassi<sub>G</sub>* e periodi di difficile comprensione;
- Controllo di formattazione IAT<sub>E</sub>X: compilatore IAT<sub>E</sub>X.

## 6.5 Walkthrough

Questa tecnica consiste nella lettura totale dell'oggetto di Verifica, senza presupposti, mirata a rilevare anomalie di cui non si è a conoscenza. E' una tecnica usata principalmente nelle prime fasi del progetto in quanto i membri del gruppo non possiedono ancora l'esperienza necessaria per una Verifica più mirata. Verrà utilizzata dal gruppo in prima persona durante la redazione dei documenti, anche da componenti del gruppo diversi dal redattore, data la natura ancora informale del documento analizzato. Inoltre essa verrà implicitamente utilizzata da tutti gli strumenti software di analisi statica utilizzati per la Verifica. La Verifica tramite walkthrough viene effettuata in 4 fasi distinte:

1. Pianificazione della Verifica;
2. Lettura;
3. Discussione, fase in cui si riportano gli errori riscontrati, al fine di rendere più specifica la ricerca;
4. Correzione dei difetti.

## 6.6 Inspection

Questa tecnica consiste nella ricerca di anomalie velocizzata da assunzioni riguardanti la tipologia e la posizione sulla base di conoscenze di tipo statistico, apprese dai Verificatori attraverso la tecnica del walkthrough. Grazie a queste assunzioni l'inspection è una strategia più rapida del walkthrough e non necessita della lettura integrale del documento in oggetto. Ogni applicazione della tecnica di inspection si divide in quattro fasi distinte:

1. Pianificazione della Verifica;
2. Definizione della lista di controllo, ovvero il riferimento ad un preciso elenco di Verifica mirata;
3. Lettura;
4. Correzione.

Durante l'applicazione del walkthrough 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.
- $\text{IAT}_{\text{E}}\text{X}$ :
  - Lettere accentate nelle variabili: non viene utilizzato il comando apposito;
  - Carattere di spaziatura: non deve essere utilizzato all'interno dei tag;
  - Macro  $\text{IAT}_{\text{E}}\text{X}$ : non viene scritta usando l'apposito comando  $\text{IAT}_{\text{E}}\text{X}$ .
- 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.

Per quanto riguarda la Verifica mirata del codice, non abbiamo sviluppato un elenco di errori comuni, in quanto non compete alla fase di Analisi dei Requisiti la stesura di codice. Quindi ci appoggiamo su tecniche di *best practice<sub>G</sub>* associate per avere un insieme di ispezioni di riferimento. Queste tecniche comprendono il controllo in più campi:

- Errori nei dati:
  - Le variabili sono state inizializzate prima di essere utilizzate?;
  - Le costanti sono state specificate e nominate correttamente?;
  - Gli *upper bound<sub>G</sub>*, nello scorrimento di un *array<sub>G</sub>*, sono corretti?;
  - Esiste la possibilità del verificarsi di un *overflow<sub>G</sub>*?
- Errori nei controlli:
  - Le condizioni negli *statement<sub>G</sub>* condizionali sono effettivamente corrette?;
  - Ogni ciclo è certo di terminare?;
- Errori input/output:
  - Le variabili di *input<sub>G</sub>* sono state interamente utilizzate?;
  - E' stato assegnato un valore ad ogni variabile di *output<sub>G</sub>* prima di essere esportata?;
  - Input imprevisti possono causare corruzione?.
- Errori di *interfaccia<sub>G</sub>*:
  - Tutte le funzioni e le chiamate di metodi hanno il corretto numero di parametri?;
  - I tipi dei parametri formali e attuali sono gli stessi?;
  - Tutti i parametri sono nell'ordine giusto?;
  - Se i componenti condividono della memoria, condividono anche la struttura di memorizzazione condivisa?.
- Errori di gestione della memoria:



- Se una struttura a riferimenti viene modificata, tutti i riferimenti sono stati correttamente riassegnati?;
  - Se viene usata un'allocazione dinamica della memoria, lo spazio viene allocato correttamente?;
  - Lo spazio occupato viene esplicitamente deallocato, una volta che non è più richiesto?.
- Errori di gestione delle eccezioni:
    - Tutte le possibili eccezioni vengono tracciate e gestite?.

### **6.7 Calcolo indice Gulpease**

Il calcolo dell'indice di Gulpease dovrà essere applicato dopo l'attività di verifica dello stesso documento, ma prima dell'approvazione. Il risultato di questo calcolo sarà un indice della qualità del documento. Se l'indice si scosterà in negativo dagli obiettivi del gruppo, si dovrà informare il Responsabile. Il Responsabile, successivamente, potrà decidere di chiedere una revisione dell'intero documento al fine di aumentarne la leggibilità.