

SWEENEYTHREADS

ACTORBASE

A NoSQL DB BASED ON THE ACTOR MODEL

Norme di progetto

Redattori:

Nicoletti Luca

Tommasin Davide

Approvazione:

Verifica:



Versione 1.3.3

21 marzo 2016

Indice

1	Introduzione	3
1.1	Scopo del documento	3
1.2	Scopo del prodotto	3
1.3	Glossario	3
1.4	Riferimenti	4
1.4.1	Informativi	4
1.4.2	Normativi	4
2	Processi primari	5
2.1	Fornitura	5
2.1.1	Studio di fattibilità	5
2.1.2	Casi d'uso	5
2.1.3	Tecniche di tracciamento dei requisiti	6
2.1.4	Gestione cambiamento requisiti	6
2.2	Sviluppo	6
2.2.1	Codifica e convenzioni	6
2.2.2	L ^A T _E X	6
2.2.3	Scala	7
3	Processi di supporto	8
3.1	Tecniche di analisi e classificazione requisiti	8
3.2	Documentazione	8
3.2.1	Template	8
3.2.2	Struttura documenti	8
3.2.3	Prima pagina	9
3.2.4	Indice	9
3.2.5	Diario delle modifiche	9
3.2.6	Formattazione generale delle pagine	10
3.2.7	Norme tipografiche	10
3.2.8	Stile del testo	10
3.2.9	Formati	10
3.2.10	Sigle	11
3.2.11	Componenti grafiche	11
3.2.12	Classificazione documenti	12
3.2.13	Versionamento documenti	12
3.2.14	Ciclo di vita dei documenti	12
4	Processi organizzativi	14
4.1	Processi di gestione dell'infrastruttura	14
4.1.1	Documentazione di pianificazione	14
4.1.2	Ticketing	14
4.1.3	Versioning	15
4.1.4	Repository	16
4.2	Processi di management	17
4.2.1	Ruoli	17
4.2.2	Comunicazioni	17
4.2.3	Riunioni	18
	Elenco delle figure	19
	Elenco delle tabelle	20

Diario delle modifiche

Versione	Data	Autore	Descrizione
1.3.3	2016-03-21	<i>Analista</i> Tommasin Davide	Controllo e sostituzione di termini all'interno della sezione di introduzione del documento
1.3.2	2016-03-11	<i>Progettista</i> Nicoletti Luca	Inserita in Sezione 3.1.3 la norma riguardante i nomi in prima pagina di ogni documento. Inserito in Sezione 3.1.14 una norma riguardante il controllo ortografico prima di sottomettere i documenti all'attività di verifica. Spostata vecchia sezione 2.1.1 in Sezione 3.1 Tecniche di analisi e classificazione dei requisiti.
1.3.1	2016-03-01	<i>Responsabile</i> Nicoletti Luca	Modifiche apportate a tutti gli errori riscontrati dall'analisi effettuata in data odierna. Inserita norma per nuovi comandi per scrivere riferimenti a dei documenti o a dei termini di glossario in sezione 2.2.2.
1.3.0	2016-02-23	<i>Responsabile</i> Nicoletti Luca	Prime modifiche post revisione RR: riviste le norme riguardanti il versionamento dei documenti in sezione 3.1.13; modificata l'intera tabella del diario delle modifiche; inserita norma riguardante i termini presenti nel glossario per ogni documento.
1.2.1	2016-01-18	<i>Responsabile</i> Padovan Tommasin	Documento approvato, pronto alla consegna
1.2.0	2016-01-18	<i>Verificatore</i> Tommasin Davide	Verifica effettuata, segnalati errori e dimenticanze
1.1.2	2016-01-17	<i>Amministratore</i> Nicoletti Luca	Modifica margini, inserita norma sui margini
1.1.1	2016-01-11	<i>Amministratori</i> Maino Elia, Nicoletti Luca	Stesura norme riguardanti commit e politiche di ticketing, correzione errori
1.1.0	2016-01-10	<i>Verificatori</i> Biggeri Mattia, Tommasin Davide	Verifica del documento
1.0.5	2016-01-09	<i>Amministratore</i> Ni- coletti Luca	Stesura norme riguardanti documentazione di pianificazione, correzione errori
1.0.4	2016-01-09	<i>Amministratore</i> Maino Elia	Stesura norme riguardanti risoluzione problemi, versioning, comunicazioni e casi d'uso
1.0.3	2016-01-09	<i>Amministratore</i> Ni- coletti Luca	Stesura norme riguardanti Ticketing, Repository, Ruoli, Riunioni
1.0.2	2016-01-08	<i>Amministratore</i> Maino Elia	Stesura norme Documentazione
1.0.1	2016-01-08	<i>Amministratore</i> Ni- coletti Luca	Stesura norme riguardanti Studio di fattibilità e requisiti, codifica \LaTeX e <i>Scala</i>
1.0.0	2016-01-08	<i>Amministratori</i> Maino Elia, Nicoletti Luca	Creazione scheletro documento e struttura organizzativa: inserite le sezioni: Introduzione, Processi primari, Processi di supporto, Processi organizzativi Stesura Introduzione

Tabella 1: Diario delle modifiche

1 Introduzione

1.1 Scopo del documento

Nel seguente documento sono definite le norme che l'intero gruppo SWEeneyThreads si impegna a rispettare durante lo svolgimento del progetto Actorbase.

Ogni membro è tenuto a leggere il documento e a rispettare le norme al fine di dare maggiore uniformità allo svolgimento dei processi, migliorandone l'efficacia, riducendo il numero di errori e i tempi di sviluppo.

Poiché il gruppo ha deciso di basarsi sulla struttura a processi *ISO/IEC 12207* la struttura di questo documento ne rispecchia l'organizzazione. In particolare la suddivisione in processi primari, di supporto e organizzativi.

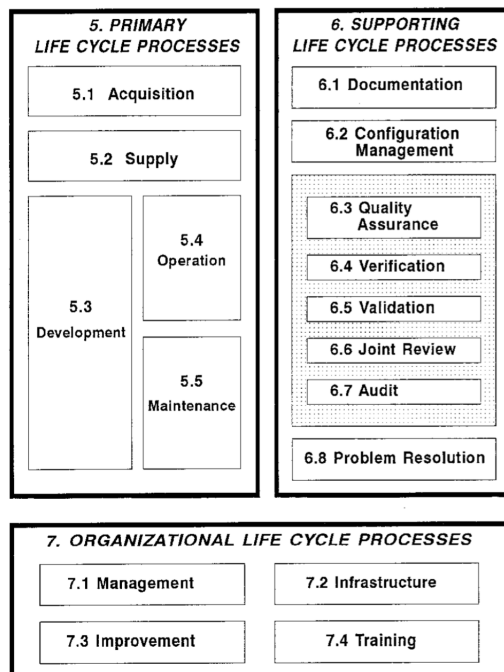


Figura 1: Processi ISO/IEC 12207

1.2 Scopo del prodotto

Lo scopo del progetto è la realizzazione di un DataBase NoSQL key-value basato sul modello ad Attori con l'obiettivo di fornire una tecnologia adatta allo sviluppo di moderne applicazioni che richiedono brevissimi tempi di risposta e che elaborano enormi quantità di dati. Lo sviluppo porterà al rilascio del software sotto licenza MIT.

1.3 Glossario

Al fine di evitare ambiguità di linguaggio e di massimizzare la comprensione dei documenti, il gruppo ha steso un documento interno che è il *Glossario v1.0.3*. In esso saranno definiti, in modo chiaro e conciso i termini che possono causare ambiguità o incomprensione del testo. In ogni documento, ogni occorrenza di un termine presente nel glossario, verrà contrassegnata con una G a pedice come la seguente:

$G.$

e per questo è stato introdotto un nuovo comando in tutti i documenti:

`\newcommand{\GLO}{\ped{\textit{G.}}}`.

1.4 Riferimenti

1.4.1 Informativi

- Specifiche UTF-8:
http://unicode.org/faq/utf_bom.html
- Licenza MIT:
<https://opensource.org/licenses/MIT>
- Scala Programming Language:
<http://www.scala-lang.org/>
- ISO/IEC 12207:
http://www.iso.org/iso/catalogue_detail?csnumber=43447
- ISO 8601:2004:
<http://www.iso.org/iso/home/standards/iso8601.htm>
- L^AT_EX:
<https://www.latex-project.org>
- UML:
<http://www.uml.org>
- StarUML:
<http://staruml.io>
- Telegram:
<https://telegram.org>
- Google Drive:
https://www.google.com/intl/it_it/drive/
- Google Hangouts:
<https://hangouts.google.com>
- TeamWork:
<https://www.teamwork.com/>
- IntelliJ:
<https://www.jetbrains.com/idea/>
- ProjectLibre:
<http://www.projectlibre.org/>
- GitHub:
<https://github.com/>
- Texmaker:
<http://www.xm1math.net/texmaker/>
- altervista:
<http://it.altervista.org/>
- Piano di progetto:
Piano di Progetto v1.0.0
- Piano di qualifica:
Piano di Qualifica v1.0.0

1.4.2 Normativi

- Capitolato d'appalto Actorbase (C1):
<http://www.math.unipd.it/~tullio/IS-1/2015/Progetto/C1p.pdf>

2 Processi primari

Sono state definite delle norme relative ai processi primari che maggiormente riguardano le attività svolte dal gruppo: fornitura e sviluppo.

2.1 Fornitura

2.1.1 Studio di fattibilità

Lo *Studio di fattibilità* del progetto deve essere steso dai membri che ricoprono il ruolo di *Analisti* sulla base delle prime riunioni effettuate, decise dal *Responsabile di progetto*, e delle preferenze espresse da ogni singolo membro del gruppo. In seguito il documento verrà analizzato e valutato da altri membri del gruppo. Lo *Studio di fattibilità* deve contenere:

- **Dominio:** conoscenza delle tecnologie richieste e del dominio applicativo;
- **Rapporto costo/benefici:** eventuali prodotti simili già presenti sul mercato, competitori, costo della realizzazione del prodotto e quantità di requisiti obbligatori;
- **Individuazione dei rischi:** evidenziare lacune tecniche e di conoscenza del dominio dei membri del gruppo, comprensione dei punti critici, difficoltà nel determinare i requisiti obbligatori e opzionali e nella loro classificazione;

2.1.2 Casi d'uso

Come detto in precedenza alcuni requisiti possono essere ricavati dai Casi d'uso_G, ad essi si può fare riferimento anche con la dicitura *use case* o con l'acronimo *UC* nel caso fosse necessario utilizzarli in tabelle o diagrammi.

I Casi d'uso_G vanno identificati dagli *Analisti*, attraverso una procedura che va dal generale al particolare.

Un *caso d'uso* richiede la definizione dei seguenti campi:

- Codice gerarchico
- Nome sintetico
- Attori
 - Principali
 - Secondari
- Pre-condizione
- Post-condizione
- Flusso degli eventi relativi allo scenario principale
- Eventuali scenari alternativi
- Lista di requisiti dedotti

A tale insieme di informazioni va associato un diagramma di caso d'uso in UML. Per la scrittura dei Casi d'uso_G in UML il gruppo ha deciso di utilizzare il software *StarUML* alla versione 2.5.1. La scelta è caduta su tale programma poiché è disponibile per tutti i principali sistemi operativi desktop e permette di utilizzare una versione di prova completa per tutto il tempo del progetto.

È degli *Analisti* il compito di inserire tutte le informazioni relative ai casi d'uso e i loro diagrammi nel documento di *Analisi dei requisiti*.

Il codice gerarchico del *caso d'uso* ha la forma seguente:

UC[codice univoco del padre].[codice progressivo]

Dove il codice progressivo può definire diversi livelli di gerarchia separati da un punto.

I Casi d'uso_G saranno espressi in forma tabellare nel documento di *Analisi dei requisiti* nel seguente modo, in più, ogni caso d'uso sarà accompagnato da un diagramma UML che ne descrive il flusso di eventi.

Codice gerarchico	UC1
Nome sintetico	Connessione al server
Attore principale	Utente non autenticato
Attori secondari	Nessuno
Pre-condizione	L'utente ha avviato l'interfaccia CLI, non è autenticato e intende connettersi ad un server
Post-condizione	L'utente risulta connesso al server
Flusso eventi	1. L'utente inserisce il comando CONNECT (UC 1.1) 2. L'utente inserisce l'indirizzo del server (UC 1.2) 3. L'utente inserisce username e password (UC 1.3) e preme invio
Scenari alternativi	Nessuno
Lista requisiti	...

Tabella 2: Caso d'uso UC 1 - Connessione al server

2.1.3 Tecniche di tracciamento dei requisiti

Tutti i requisiti e Casi d'uso_G saranno inseriti in un Database, strutturato in modo funzionale dal gruppo. Nel presente Database verranno inserite anche le *Milestones* definite dal responsabile di turno, in modo da poter collegare requisiti e *Milestones*. Verranno creati degli appositi triggers per automatizzare la validazione dei requisiti tramite completamento manuale delle *Milestones* presenti.

Il Database verrà scritto utilizzando il linguaggio SQL, sfruttando l'hosting gratuito *allevista* e utilizzando il servizio proprietario per la gestione di tale database.

Questo compito è di responsabilità dell'*Amministratore*.

2.1.4 Gestione cambiamento requisiti

Per quanto riguarda il cambiamento dei requisiti, nel Database verrà tenuta una tabella di backup dei "vecchi" requisiti, nella loro forma di dichiarazione, con un puntatore al "nuovo" requisito, che invece avrà le specifiche aggiornate. È compito del *Responsabile di progetto* mantenere aggiornata la tabella dei requisiti, copiando prima il requisito che necessita di cambiamento nella tabella dei vecchi requisiti, e poi aggiornando il requisito stesso.

Tutto questo al fine unico di avere una tracciabilità del cambiamento dei requisiti visionabile a posteriori.

2.2 Sviluppo

2.2.1 Codifica e convenzioni

Di seguito sono riportate le norme che il gruppo andrà a seguire durante la stesura di codice, qualsiasi esso sia.

2.2.2 L^AT_EX

Regole riguardanti L^AT_EX:

- Ogni file deve iniziare con 3 righe di commento come quelle riportate in seguito:

```
%Document-Author: Cognome Nome + Cognome Nome
%Document-Date: aaaa/mm/gg
%Document-Description: descrizione
```

Figura 2: L^AT_EX - Commenti ad inizio file

- Ogni file deve contenere nella prima parte tutti gli `\usepackage{}` necessari
- I commenti andranno inseriti in una riga vuota, eventualmente prima della riga di codice a cui fanno riferimento
- I commenti su più righe useranno il comando `\begin{comment}` - `\end{comment}`
- Tra ogni `\begin{PART}` e `\end{PART}` tutto il testo e il codice andrà indentato:

```
\begin{PART1}
  testo
  \begin{PART2}
    testo
    testo
    \end{PART2}
\end{PART1}
```

Figura 3: L^AT_EX - Indentazione 1

- Per quanto riguarda il comando personalizzato `\newpage` `\section{}` o altre sezione `\subsection{}`, `\subsubsection{}` verranno comunque indentate le parti innestate al loro interno come segue:

```
\mychapter{n}{Titolo}
  Testo
  \section{sezione}
    testo
    \subsection{sottosezione}
  \section{sezione2}
    testo
    \subsection{sottosezione2}
      \subsubsection{non numerata}
        testo
      \subsubsection{non numerata2}
        testo
    testo
  \section{sezione3}
```

Figura 4: L^AT_EX - Indentazione 2

- Verrà utilizzato T1 come encoding del font:
`\usepackage[T1]{fontenc}`
- Verrà utilizzato utf8 come encoding dell'input:
`\usepackage[utf8]{inputenc}`
- Verrà utilizzato english, italian come parametro per babel:
`\usepackage[english, italian]{babel}` in modo da usare inglese e italiano nello stesso documento tenendo italiano come lingua principale
- Prima di ogni immagine, verrà inserito un commento su una riga, come definito sopra, per facilitarne l'individuazione:

```
%immagine
\begin{figure}[H]
  \centering
  \includegraphics[scale=x.x]{nome_file.estensione}
  \caption{Titolo immagine}
\end{figure}
```

Figura 5: L^AT_EX - Commento prima di ogni immagine

- A fine documento, come commento su più righe, andrà inserita la documentazione e la descrizione (anche breve) del file

2.2.3 Scala

Tutte le regole di indentazione, assegnazione dei nomi, scrittura delle parentesi, nominazione file, e documentazione sono quelle definite dalla documentazione ufficiale di *Scala*: <http://docs.scala-lang.org/style/>

3 Processi di supporto

3.1 Tecniche di analisi e classificazione requisiti

Sempre compito degli *Analisti*, sarà quello di stilare l'*Analisi dei requisiti*. Essi potranno ricavarli da eventuali Casi d'uso_G emersi da Brainstorming o riunioni con il committente.

I requisiti saranno elencati secondo un ordine. Ogni requisito seguirà la seguente codifica:

R[Codice][Importanza][Tipo]

Codice

Un codice univoco ed espresso in modo gerarchico;

Importanza

Può assumere i seguenti valori:

- **N:** Necessary (obbligatorio);
- **D:** Desiderable (desiderabile, a valore aggiunto);
- **O:** Optional (opzionale).

Tipo

Può assumere i seguenti valori:

- **F:** funzionale;
- **Q:** di qualità;
- **P:** prestazionale;
- **V:** vincolo.

I requisiti saranno inseriti in una tabella, che includerà anche un nome per ogni attributo, una fonte e una breve descrizione. La forma tabellare di un requisito risulta quindi essere:

Codice	Nome	Fonte	Descrizione
R[1.4.5][N][P]	Requisito di qualità	Capitolato	Garantire ...

3.2 Documentazione

In questo capitolo si descrivono le convenzioni definite e adottate dal gruppo riguardanti le modalità di redazione, verifica e approvazione dei documenti.

Tutti i documenti formali prodotti da SWEneyThreads sono scritti utilizzando il linguaggio \LaTeX , compilati e forniti in formato PDF (per quanto riguarda le versioni digitali). Per la stesura dei documenti il gruppo utilizzerà il software *Texmaker*.

3.2.1 Template

Al fine di rendere più rapida e meno incline a differenziazioni la stesura dei diversi documenti è stato prodotto un template \LaTeX , reperibile nel repository in `Actorbase/LaTeX/Templates`.

3.2.2 Struttura documenti

La struttura dei documenti presenta una suddivisione in sezioni, sottosezioni e ulteriori sotto-sottosezioni. Tutti le sezioni, sottosezioni e sotto-sottosezioni sono state create usando i comandi standard \LaTeX `\section{}`, `\subsection{}` e `\subsubsection{}`.

La numerazione delle sezioni è utilizzata fino al terzo livello di profondità (x.y.z), dal quarto livello in poi le sottosezioni non presentano numerazione. Tale scelta è stata presa al fine di rendere più leggibile

l'indice.

Di seguito viene fornita una descrizione più dettagliata di alcuni elementi di un documento:

3.2.3 Prima pagina

La prima pagina di un documento presenta gli elementi seguenti:

- Nome del gruppo
- Nome del progetto
- Sottotitolo del progetto
- Titolo del documento
- Cognome e nome dei redattori del documento
- Cognome e nome di chi approva il progetto in qualità di responsabile
- Cognome e nome dei verificatori del documento
- Logo del gruppo
- Numero di versione del documento
- Data di rilascio del documento

La prima pagina è parte del template disponibile nel repository.

I cognomi e i nomi dei redattori, del responsabile e dei verificatori, verranno cambiati di versione in versione. Questo è possibile grazie al versionamento dei documenti, infatti, i nomi di chi lavora al documento, fanno riferimento solo alla versione in cui il documento è in un determinato momento. Questo comporta un cambiamento dei nomi dei responsabili di un documento di versione in versione. Il numero di nomi è quindi variabile, in una fase ci possono essere 2 persone per la scrittura del documento, in quella successiva 3, o solamente una.

3.2.4 Indice

In ogni documento sono presenti in ordine

- Un indice delle sezioni;
- Un indice delle tabelle;
- Un indice delle figure.

Tali indici sono generati automaticamente tramite appositi comandi \LaTeX , l'assenza di figure e/o tabelle nel documento comporta l'omissione del corrispondente indice.

Data la natura secondaria degli indici relativi alle tabelle e alle figure, si è deciso di posizionarli alla fine del documento. L'indice dei contenuti si trova invece subito dopo la pagina iniziale.

3.2.5 Diario delle modifiche

Ogni documento deve contenere una sezione denominata "Diario delle modifiche" in cui annotare tutte le attività svolte sul documento.

Lo schema della tabella è il seguente:

- **Versione:** numero di versione del documento dopo le modifiche;
- **Data:** data in cui sono state apportate le modifiche;
- **Autore:** ruolo, cognome e nome dell'autore che apportato le modifiche (gli autori possono essere più di uno);
- **Descrizione:** descrizione delle modifiche apportate al documento.

La compilazione della tabella è un'attività obbligatoria nel caso di modifiche rilevanti al documento. Un documento non può cambiare di versione senza che tale cambiamento venga annotato nella tabella, nel caso di modifiche minori che non cambiano di molto il contenuto (es. correggere un accento) la tabella può rimanere invariata.

Versione	Data	Autore	Descrizione
----------	------	--------	-------------

Tabella 3: Schema del diario delle modifiche

Il Diario delle modifiche non è incluso nella numerazione delle sezioni, si trova dopo l'indice e prima di qualsiasi capitolo numerato

3.2.6 Formattazione generale delle pagine

La formattazione generale di una pagina prevede la diminuzione dei margini destri e sinistri, ma non prevede altre modifiche importanti e si basa, per tutte le altre regole, sulla formattazione standard di \LaTeX usata per i documenti di classe "Report".

Per effettuare la modifica viene usato i seguenti comandi:

```
\usepackage{geometry}
\geometry{margin=1in}.
```

3.2.7 Norme tipografiche

Questa sezione contiene norme tipografiche e ortografiche adottate dal gruppo al fine di garantire uno stile uniforme e una semantica coerente per tutti i documenti.

3.2.8 Stile del testo

Il font utilizzato in tutti i documenti formali scritti dal gruppo sarà il *Computer Modern*, ovvero quello standard utilizzato da \LaTeX .

- **Corsivo:** il corsivo va utilizzato nei casi seguenti:

- Citazioni;
- Nomi particolari;
- Documenti;
- Riferimenti;

A seconda della semantica del testo si utilizzano i comandi \LaTeX `\emph{}` e `\textit{}`.

- **Grassetto:** il grassetto va utilizzato nei casi seguenti:

- Elenchi puntati: evidenzia il concetto sviluppato nella continuazione del punto.

- **Maiuscolo:** una parola completamente in maiuscolo deve indicare un acronimo o una sigla.

- **\LaTeX :** ogni riferimento al linguaggio \LaTeX va scritto utilizzando il comando `\LaTeX`.

3.2.9 Formati

- **Percorsi:**

- Indirizzi email : comando \LaTeX `\href{mailto:nome@dominio}{nome@dominio}`;
- Indirizzi web completi: comando \LaTeX `\url`;
- Indirizzi relativi: comando \LaTeX `\verb`.

- **Date:** le date presenti nei documenti seguono lo standard ISO 8601:2004:

AAAA - MM - GG

Dove:

- AAAA rappresenta l'anno;
 - MM rappresenta il mese;
 - GG rappresenta il giorno.
- **Ruoli di progetto:** quando si fa riferimento ad un ruolo di progetto questo va scritto in corsivo e con la prima lettera maiuscola (es. *Responsabile*).
 - **Documenti:** i riferimenti vanno scritti in corsivo (es. *Analisi dei requisiti*).
 - **Nomi dei file:** i nomi dei file vanno scritti utilizzando il comando `\verb` (es. `immagine.png`).
 - **Nomi propri:** I nomi propri seguono la forma "Cognome Nome".
 - **Nome del gruppo:** il nome del gruppo è SWEneyThreads, la distinzione tra lettere maiuscole e minuscole va rispettata ogni volta che vi si fa riferimento.

3.2.10 Sigle

L'utilizzo di sigle e abbreviazioni per riferirsi a documenti va limitato il più possibile, tuttavia nel caso il loro uso fosse funzionale alla lettura (come nel caso di tabelle o diagrammi) il loro uso è consentito:

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

3.2.11 Componenti grafiche

Le componenti grafiche previste all'interno dei documenti sono immagini e tabelle. Ogni occorrenza di un elemento grafico è accompagnata da una didascalia indicizzata, in modo da poterla associare alla sezione relativa del documento. **Tabelle** Le tabelle sono definite utilizzando un template in `LaTeX` realizzato dal gruppo e disponibile nel repository all'indirizzo `Actorbase/LaTeX/Templates` **Immagini** Il formato scelto per le immagini è Portable Network Graphics (PNG).

Le immagini vanno sempre inserite utilizzando la seguente sequenza di comandi `LaTeX`:

```
\begin{figure}[H]
\centering
\includegraphics[scale=0-1]{Immagini/nome.png}
\caption{Titolo - didascalia}
\end{figure}
```

3.2.12 Classificazione documenti

I documenti prodotti dal gruppo si dividono in formali e informali. **Documenti formali** Quando un documento riceve l'approvazione del *Responsabile* viene definito formale e risulta idoneo al rilascio all'esterno del gruppo.

Per risultare approvato un documento deve aver completato con successo il percorso di verifica e validazione descritto nel *Piano di Qualifica*. **Documenti informali** Un documento rimane informale finché non viene approvato dal *Responsabile*, durante tale fase il suo uso è da considerarsi esclusivamente interno al gruppo.

Alcuni documenti prodotti dal gruppo possono rimanere informali per l'intera durata del loro ciclo di vita.

3.2.13 Versionamento documenti

I documenti prodotti dal gruppo devono essere sempre identificati da un numero di versione del tipo:

X.Y.Z

Dove:

- X: è il numero principale di versione, viene incrementato ad ogni uscita formale del documento;
- Y: viene incrementato quando il documento entra in una fase successiva del suo ciclo di vita;
- Z: viene incrementato quando si apportano modifiche minori al documento.

All'interno di un documento quando si intende fare riferimento ad una specifica versione di un altro documento la notazione da utilizzare è:

Nome Documento vX.Y.Z.

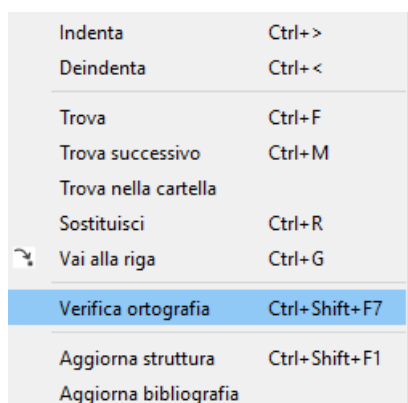
Mentre per fare riferimento ad un file vero e proprio:

NomeDocumento_vX.Y.Z.estensione

3.2.14 Ciclo di vita dei documenti

Ogni documento prodotto dal gruppo rispetta il seguente ciclo di vita:

- **Lavorazione/Modifica:** il documento entra in questa fase al momento della sua creazione e vi rimane per tutto il tempo in cui il suo contenuto viene modificato. Prima di terminare la sua fase di modifica ed essere messo a disposizione dei verificatori, su ogni documento deve essere effettuato il controllo ortografico messo a disposizione dal software *TexMaKer*:



Indenta	Ctrl+>
Deindenta	Ctrl+<
Trova	Ctrl+F
Trova successivo	Ctrl+M
Trova nella cartella	
Sostituisci	Ctrl+R
Vai alla riga	Ctrl+G
Verifica ortografia	Ctrl+Shift+F7
Aggiorna struttura	Ctrl+Shift+F1
Aggiorna bibliografia	

Figura 6: Controllo ortografico - strumento di TexMaKer

- **Verifica:** quando termina la fase di modifica, il documento passa nelle mani dei *Verificatori* che lo analizzano al fine di individuare eventuali errori o incongruenze sintattiche e semantiche;

- **Approvazione:** dopo essere stato verificato il documento deve essere approvato dal *Responsabile*. Se il documento ottiene l'approvazione diventa ufficiale e raggiunge lo stato finale del suo ciclo di vita per quanto riguarda la corrente versione.

Ogni documento prodotto può attraversare più volte ogni fase del suo ciclo di vita, allo stesso modo può non attraversarle tutte. Quando si inizia una revisione formale su un documento già approvato questo ricomincia il ciclo da capo con un numero di versione incrementato.

4 Processi organizzativi

4.1 Processi di gestione dell'infrastruttura

4.1.1 Documentazione di pianificazione

Per quanto riguarda la documentazione della pianificazione si è scelto di adoperare *ProjectLibre*, un software Open-source per il project management, che permette di automatizzare molte mansioni che altrimenti il *Progettista* dovrebbe svolgere a mano.

ProjectLibre è stato scelto per le sue ottime caratteristiche:

- Portabilità, in quanto basato su *Java*;
- Open-source;
- Genera automaticamente diagrammi di Gantt, WBS e PERT;
- Calcola automaticamente i costi, sia totali che per singola attività/risorsa, aiutando a tenere sotto controllo il budget.

4.1.2 Ticketing

Scelta della piattaforma di ticketing Per quanto riguarda l'emissione e la gestione dei ticket si è scelto di affidarsi alla piattaforma *Teamwork* in quanto:

- Ha ottenuto buoni punteggi da reviews di utenti e di critica;
- Fornisce 100Mb di storage e la possibilità di avere due progetti attivi, contemporaneamente;
- Fornisce un analizzatore di rischi e benefici;
- Genera automaticamente diagrammi di Gantt interattivi;
- Include un ottimo Task management (priorità, task history, possibilità di aggiungere in automatico task ricorrenti);
- Notifiche sms e *Notification group*.

La principale alternativa presa in considerazione è stata *Zoho*, ma non è stata ritenuta all'altezza in quanto offre meno features. Segue una breve lista per mettere a confronto le principali funzionalità messe a disposizione dalle due piattaforme:

ZOHO	TEAMWORK
Calendar	Calendar
Gant	Gantt interattivi
Task management	To-do list
Time tracking	Track Project Hours
Bug tracking	Analizzatore rischi/benefici
Document management	Template di progetto
	Priorities
	Track Burn Rate
	Track Staff Hours
	SMS di notifica

Tabella 4: Zoho / Teamwork - Lista features

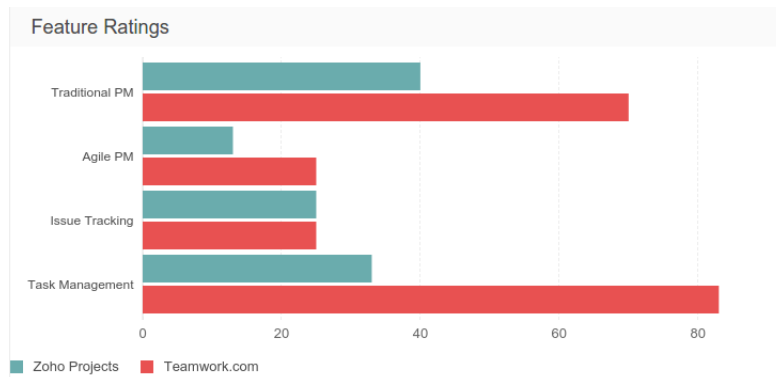


Figura 7: Zoho / Teamwork - Rating delle features a confronto

Secondo *SoftwareInsider* (softwareinsider.com) sono molto simili nelle funzionalità principali; ma *Teamwork* offre alcuni strumenti in più per la gestione di processi software tradizionali.

I principali:

	ZOHO	TEAMWORK
Calendar	✓	✓
Gantt interattivi	✓	✓
Template di progetto	✓	✓
Risk/benefits analyzer	✗	✓
Scheduling	✗	✓

Tabella 5: Zoho / Teamwork - Differenza strumenti

Come task management *Zoho* offre solamente delle To-do List, mentre *Teamwork* ha anche le seguenti feature:

- Add Recurring Tasks;
- Group Tasks by Projects;
- Set Priorities;
- Task History.

Zoho offre alcune funzionalità in più in quanto a comunicazione real-time tra membri del gruppo, ma questo risulta irrilevante per il nostro gruppo, in quanto per la comunicazione real-time viene adottato un sistema diverso. **Politiche di ticketing** I Ticket devono essere assegnati dal *Responsabile*. Poiché un ticket assegna una o più attività ad una o più persone, chi riceve un ticket può generare altri ticket relativi a sotto attività da lui individuate per tenere traccia dello sviluppo in maniera più chiara. In ogni caso a tutti gli altri membri del gruppo non è assolutamente permessa la generazione di ticket esterni ad attività già assegnate.

4.1.3 Versioning

Per gestire il versionamento il gruppo utilizza *GitHub*. Tale scelta è dovuta sia ad un apprezzamento comune da parte dei membri del gruppo per la piattaforma, che ad una richiesta esplicita di pubblicazione del progetto sulla stessa da parte del committente.

È stato creato un account ufficiale del gruppo, raggiungibile all'indirizzo <https://github.com/SweeneyThreads>

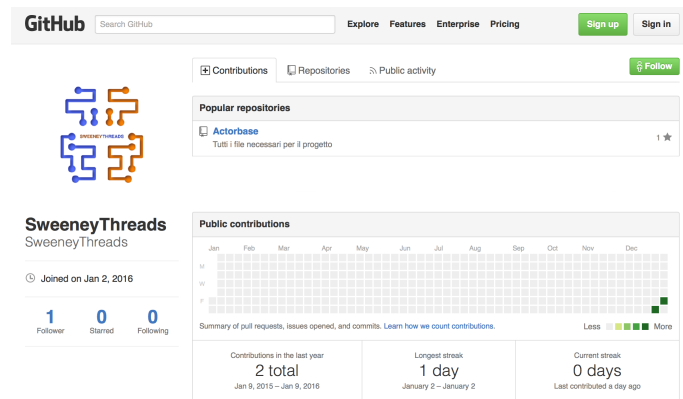


Figura 8: Account GitHub SWEeneyThreads

4.1.4 Repository

Sono state previste diverse *Repository* necessarie allo sviluppo del progetto: Actorbase, RR, RP, RQ, RA. Actorbase conterrà tutti i file del prodotto da sviluppare, mentre le *Repository* RR, RP, RQ e RA si riferiscono alle 4 consegne del progetto previste: revisione dei requisiti, revisione di progettazione, revisione di qualifica, revisione di accettazione. Effettuata la consegna del materiale, la *Repository* Actorbase/ verrà copiata in quella corrispondente, che servirà quindi come backup della *Baseline* a cui fa riferimento.

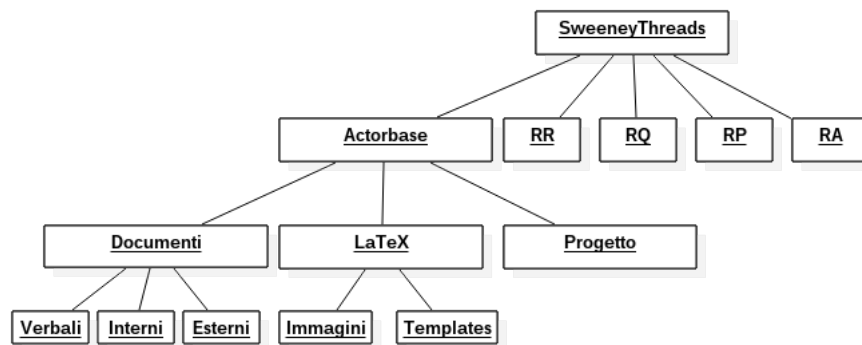


Figura 9: Struttura dei repositories GitHub

In Actorbase/ saranno presenti le seguenti sottocartelle:

- Documenti;
- LaTeX;
- Progetto.

Documenti Nella cartella Actorbase/Documenti/ verranno inseriti tutti i pdf generati dal comando `pdflatex nome-documento.tex`. Non saranno presenti altri file in questa cartella. I documenti saranno divisi in *Interni*, *Esterni* e *Verbali* per questo saranno create delle sottodirectory: Actorbase/Documenti/Interni, Actorbase/Documenti/Esterni e Actorbase/Documenti/Verbali. **LaTeX** Nella cartella Actorbase/LaTeX/ saranno presenti tutti i file `*.tex` pronti per la compilazione. In questa cartella verrà inserita anche una cartella

Actorbase/LaTeX/Immagini/ contenente tutte le immagini necessarie alla compilazione dei file. Inoltre verrà aggiunta una cartella Actorbase/LaTeX/Templates/ contenente i templates per la stesura di documenti e per il disegno appropriato di tabelle. **Progetto** La cartella Actorbase/Progetto/ contiene tutti i file che compongono il prodotto richiesto nel capitolato. Sarà un progetto in *Scala* che seguirà gli standard sopra citati. **Normative per i commit** Si è deciso di dare a tutti i membri del gruppo la

possibilità di effettuare commit sul master-branch del repository senza dover attendere l'approvazione di un account centrale. Tale scelta impone però la definizione di alcune norme di commit da rispettare:

- I commit devono seguire l'emissione di un ticket. Tale norma serve ad evitare un eccessivo numero di commit sul repository contenenti poche modifiche;
- Nel caso il commit interessasse un documento o un file, il numero di versione dello stesso deve risultare aggiornato;
- Nella descrizione del commit è obbligatorio inserire una descrizione delle modifiche effettuate. È preferito l'uso di elenchi puntati, la forma discorsiva va utilizzata solo se non è possibile esprimere il contenuto come elenco.

4.2 Processi di management

4.2.1 Ruoli

Per quanto riguarda i ruoli, il gruppo utilizzerà quelli definiti nelle slide 7-11 disponibili all'indirizzo: <http://www.math.unipd.it/~tullio/IS-1/2015/Dispense/L04.pdf>. È stato deciso, unanimemente, che le rotazioni dei ruoli principali come *Amministratore* e *Responsabile di progetto* avverranno ogni 2 settimane. Come stabilito, una persona può ricoprire contemporaneamente più ruoli, la rotazione di altri ruoli come *Analista*, *Progettista* e *Programmatore* potrà avvenire meno frequentemente, in quanto potrebbe risultare dannoso dover abbandonare un'attività di analisi o di programmazione prima della sua conclusione.

Il gruppo stabilisce la rotazione dei ruoli in base alle attività, ai task, e alla disponibilità fornita da ogni membro.

È possibile ottenere in qualsiasi momento una panoramica dei ruoli assegnati tramite l'apposita sezione del progetto creato su *Teamwork*, all'indirizzo

<https://actorbase.teamwork.com/projects/188894/projectroles>.

4.2.2 Comunicazioni

Le comunicazioni possono avvenire tra membri del gruppo (interne), o tra il gruppo e terzi (esterne). Gli strumenti utilizzati differiscono a seconda della tipologia della comunicazione. **Interne**

- **Chat:** Per le comunicazioni interne il gruppo ha deciso di adottare una chat di messaggistica istantanea: *Telegram*. All'interno di questo mezzo di comunicazione verranno concordate date e orari delle riunioni; comunicati eventuali ritardi ai meeting; proposte idee informali, che verranno poi riproposte in modo ufficiale alle riunioni (questo per evitare di dimenticarsene o per lasciare tempo agli altri membri del gruppo di ragionare più tempo su una proposta); inoltre *Telegram* verrà utilizzato per l'invio di files temporanei, di documentazione o informativi. Sarà compito del *Responsabile di progetto* prelevare file di documentazione e riportarli nella repository adatta, e nel Drive del gruppo.

La scelta di *Telegram* è dovuta alla possibilità di utilizzare il servizio sia da desktop che da mobile, e alla possibilità di inviare qualsiasi tipo di file;

- **Videoconferenze:** Per le videoconferenze di gruppo si utilizzerà *Google Hangouts*.

È utilizzabile da tutti i dispositivi e richiede semplicemente un account *Google* di cui disponevano già tutti i membri del gruppo.

Esterne Per tutte le comunicazioni esterne va utilizzata la mail ufficiale del gruppo: sweeneythreads@gmail.com.

La gestione di tale indirizzo email spetta al *Responsabile* che dunque risulta essere l'unico componente del gruppo a poter comunicare con il committente in maniera ufficiale. Il *Responsabile* ha il compito di informare gli altri membri del gruppo sulle discussioni avute con il committente, tale aggiornamento può avvenire a voce durante le riunioni e gli incontri oppure tramite l'inoltro delle email ricevute agli indirizzi personali dei componenti interessati.

Le email ufficiali devono rispettare le seguenti linee guida:

- **Destinatario:** poiché questo indirizzo email va usato esclusivamente per comunicazioni ufficiali il destinatario del messaggio va salvato tra i contatti (funzione di Gmail), nel caso non dovesse già farne parte;

- **Oggetto:** l'oggetto deve esprimere in maniera chiara ed esaustiva il contenuto dell'email, deve essere breve e non deve rendere l'email confondibile con le altre preesistenti. Nel caso il messaggio fosse una risposta l'oggetto deve essere preceduto dalla particella "Re:", nel caso di un inoltro dalla particella "I:";
- **Corpo:** nel caso il messaggio fosse una risposta o un inoltro, il contenuto aggiunto va sempre scritto in testa al fine di non costringere i lettori a scorrere tutta l'email. La cancellazione della restante parte del messaggio è sconsigliata, per facilitare una visione completa della conversazione;
- **Allegati:** L'aggiunta di allegati al messaggio è consentita con l'unico vincolo di inviare file che possiedono un nome esplicativo o di specificare il contenuto dell'allegato nel corpo se il nome del file potrebbe essere poco comprensibile.

4.2.3 Riunioni

Ufficiali

Le riunioni sono divise in due: con o senza presenza del committente. Il gruppo si impegna a tenere almeno una riunione ufficiale senza presenza del committente ogni due settimane. Le riunioni hanno una durata minima di due ore, che potrà essere prolungata a piacere, in questo caso, nel verbale di riunione dovrà comparire di quanto si è superato il tempo previsto durante la riunione, e il motivo del prolungamento. Queste modifiche sono a carico del *Responsabile di progetto*. I verbali prodotti andranno inseriti nella *Repository Actorbase/Documenti/Verbali*, che verrà suddivisa in due sottocartelle per i verbali interni e quelli esterni.

Le riunioni con presenza del committente, andranno concordate secondo le norme di comunicazioni esterne con quest'ultimo e comunicate tramite i mezzi di comunicazione interni a tutti i membri del gruppo, ognuno dei quali è fortemente tenuto ad essere presente. Potranno verificarsi casi in cui non tutti i membri del gruppo potranno presentarsi alle riunioni con presenza del committente, ma non potrà verificarsi l'assenza del *Responsabile di progetto* e dell'*Amministratore*. I quali sono tenuti a riferire quanto emerso dalle riunioni a tutti i restanti membri assenti. Nelle riunioni con il committente può verificarsi un cambiamento riguardante ad un requisito, in questo caso il cambiamento va inserito nel verbale, che deve essere messo a disposizione del committente nella cartella *Actorbase/Documenti/Verbali/Esterni*.

Non ufficiali

Le riunioni non ufficiali sono da considerarsi riunioni tra pochi membri del gruppo, ad esempio tra i due realizzatori di questo stesso documento, o incontri occasionali avvenuti senza comunicazioni nei canali ufficiali. Queste riunioni non necessitano di una stesura di un verbale; se da queste riunioni emergesse un grave errore, o una comunicazione importante, i membri presenti sono tenuti a richiedere una riunione ufficiale straordinaria, che dovrà essere approvata dall'*Amministratore*. In caso contrario, tutte le scelte non rilevanti non necessitano di approvazione. **Brainstorming**

I *Brainstorming* vengono tenuti sotto richiesta di qualsiasi membro del gruppo, e approvati, se per motivazioni valide, dal *Responsabile di progetto*. Un *Brainstorming* ha durata minima di un'ora e massima di due; durante il quale ogni membro ricopre un ruolo di egual importanza rispetto agli altri, le decisioni vengono prese all'unisono o con la maggioranza dei membri a favore, non è compito del *Responsabile di progetto* approvare le soluzioni emerse da un *Brainstorming*.

Durante un *Brainstorming* ci sarà un membro con il compito di scrivere le *Minute*, ovvero un *Notaio*. Ad ogni *Brainstorming* sarà anche scelto un *Moderatore* che ricoprirà un ruolo di servizio. Ovvero dovrà far rispettare le regole di base. Una volta finito il *Brainstorming*, il *Notaio* dovrà riorganizzare gli appunti presi in un verbale ordinato.

Elenco delle figure

1	Processi ISO/IEC 12207	3
2	L ^A T _E X- Commenti ad inizio file	6
3	L ^A T _E X- Indentazione 1	7
4	L ^A T _E X- Indentazione 2	7
5	L ^A T _E X- Commento prima di ogni immagine	7
6	Controllo ortografico - strumento di TexMaKer	12
7	Zoho / Teamwork - Rating delle features a confronto	15
8	Account GitHub SWEeneyThreads	16
9	Struttura dei repositories GitHub	16

Elenco delle tabelle

1	Diario delle modifiche	2
2	Caso d'uso UC 1 - Connessione al server	6
3	Schema del diario delle modifiche	10
4	Zoho / Teamwork - Lista features	14
5	Zoho / Teamwork - Differenza strumenti	15