



AtAVi

Norme di progetto v1.0.0

Sommario

Questo documento specifica e descrive strumenti, regole e convenzioni utilizzate dal gruppo Co.Code nel corso della realizzazione del *progetto_g* AtAVi.

Versione	1.0.0
Data di redazione	2016-12-19
Redazione	Mattia Bottaro Mauro Carlin
Verifica	Luca Bertolini
Approvazione	Simeone Pizzi
Uso	Interno
Distribuzione	prof. Tullio Vardanega prof. Riccardo Cardin Co.Code

Diario delle modifiche

Versione	Riepilogo	Autore	Ruolo	Data
1.0.0	Approvazione documento	Simeone Pizzi	<i>Responsabile</i>	2016-12-19
0.3.0	Verifica intero documento	Luca Bertolini	<i>Verificatore</i>	2016-12-18
0.2.2	Completata l'intera stesura del documento	Mauro Carlin	<i>Amministratore</i>	2016-12-18
0.2.1	Correzione problemi rilevati durante la fase di verifica	Mattia Bottaro	<i>Amministratore</i>	2016-12-15
0.2.0	Verifica del documento	Luca Bertolini	<i>Verificatore</i>	2016-12-15
0.1.3	Completata stesura della sezione verifica dei processi di supporto	Mauro Carlin	<i>Amministratore</i>	2016-12-14
0.1.2	Completata stesura processi organizzativi	Mattia Bottaro	<i>Amministratore</i>	2016-12-14
0.1.1	Correzione dei problemi rilevati nella fase di verifica	Mauro Carlin	<i>Amministratore</i>	2016-12-13
0.1.0	Verifica del documento	Luca Bertolini	<i>Verificatore</i>	2016-12-13
0.0.3	Completata stesura della sezione documentazione dei processi di supporto	Mauro Carlin	<i>Amministratore</i>	2016-12-12
0.0.2	Completata stesura processo di sviluppo	Mattia Bottaro	<i>Amministratore</i>	2016-12-10
0.0.1	Inizio stesura documento	Mattia Bottaro	<i>Amministratore</i>	2016-12-10

Indice

1	Introduzione	6
1.1	Scopo del documento	6
1.2	Scopo del <i>prodotto_g</i>	6
1.3	Glossario	6
1.4	Riferimenti	6
1.4.1	Riferimenti Normativi	6
1.4.2	Riferimenti Informativi	7
2	Processi primari	8
2.1	Sviluppo	8
2.1.1	Scopo	8
2.1.2	Aspettative	8
2.1.3	Descrizione	8
2.1.4	Analisi dei requisiti	8
2.1.4.1	Scopo dell'attività	8
2.1.4.2	Aspettative dell'attività	8
2.1.4.3	Descrizione dell'attività	8
2.1.4.4	Studio di fattibilità	9
2.1.4.5	<i>Casi d'uso_g</i>	9
2.1.4.6	Codice identificativo dei <i>casi d'uso_g</i>	9
2.1.4.7	Requisiti	9
2.1.4.8	Codice identificativo dei requisiti	10
2.1.4.9	<i>UML_g</i>	10
2.1.5	Progettazione	10
2.1.5.1	Scopo dell'attività	10
2.1.5.2	Aspettative dell'attività	10
2.1.5.3	Descrizione dell'attività	10
2.1.5.4	Specifica tecnica	11
2.1.5.5	Definizione di <i>prodotto_g</i>	11
2.1.6	Codifica	11
2.1.6.1	Scopo dell'attività	11
2.1.6.2	Aspettative dell'attività	11
2.1.6.3	Descrizione dell'attività	11
2.1.6.4	Stile	11
2.1.6.5	Versionamento	12
2.1.6.6	<i>Ricorsione_g</i>	12
2.1.7	Strumenti	12
2.1.7.1	PragmaDB	12
2.1.7.2	<i>Astah_g</i>	12
3	Processi di supporto	13
3.1	Documentazione	13
3.1.1	Scopo	13
3.1.2	Aspettative	13
3.1.3	Descrizione	13
3.1.4	Procedure	13
3.1.4.1	Approvazione dei documenti	13
3.1.5	Template	13
3.1.6	Struttura dei documenti	14
3.1.6.1	Frontespizio	14
3.1.6.2	Diario delle modifiche	15
3.1.6.3	Indice	15
3.1.6.4	Intestazione e piè di pagina	15

3.1.7	Versionamento	15
3.1.8	Norme tipografiche	16
3.1.8.1	Stile del testo	16
3.1.8.2	Elenchi puntati	16
3.1.8.3	Formati comuni	16
3.1.8.4	Sigle	17
3.1.9	Elementi grafici	17
3.1.9.1	Tabelle	17
3.1.9.2	Immagini	17
3.1.10	Classificazione dei documenti	18
3.1.10.1	Documenti informali	18
3.1.10.2	Documenti formali	18
3.1.10.3	Glossario	18
3.1.10.4	Verbalì	18
3.1.11	Strumenti	19
3.1.11.1	L ^A T _E X	19
3.1.11.2	Texmaker _g	19
3.1.11.3	Excel	19
3.2	Verifica _g	20
3.2.1	Scopo del processo	20
3.2.2	Aspettative del processo	20
3.2.3	Attività	20
3.2.3.1	Analisi statica	20
3.2.3.2	Analisi dinamica	21
3.3	Qualità	21
3.3.1	Notazione	21
3.3.1.1	Metriche	21
3.3.1.2	Obiettivi	21
3.3.2	Definizione metriche	22
3.3.2.1	Qualità di processo	22
3.3.2.1.1	Percentuale di accessi avvenuti correttamente a PragmaDB - MPC1	22
3.3.2.1.2	Schedule Variance - MPC2	22
3.3.2.1.3	Cost Variance in percentuale - MPC3	22
3.3.2.1.4	Indice dei rischi non preventivati - MPC4	22
3.3.2.1.5	Numero di requisiti obbligatori soddisfatti - MPC5	23
3.3.2.1.6	Numero di requisiti desiderabili soddisfatti - MPC6	23
3.3.2.1.7	Numero di requisiti opzionali soddisfatti - MPC7	23
3.3.2.1.8	SF-IN - MPC8	23
3.3.2.1.9	SF-OUT - MPC9	23
3.3.2.1.10	???? - MPC10	23
3.3.2.1.11	Numero di metodi per classe - MPC11	23
3.3.2.1.12	Numero di parametri per metodo - MPC12	23
3.3.2.1.13	Indice di complessità ciclomatica - MPC13	23
3.3.2.1.14	Numero di livelli di annidamento - MPC14	24
3.3.2.1.15	Percentuale di linee di commento per linee di codice - MPC15	24
3.3.2.1.16	Halstead Difficulty per funzione - MPC16	24
3.3.2.1.17	Halstead Volume per funzione - MPC17	24
3.3.2.1.18	Halstead Effort per funzione - MPC18	24
3.3.2.1.19	Indice di manutenibilità - MPC19	25
3.3.2.1.20	Percentuale di componenti integrate nel sistema - MPC20	25
3.3.2.1.21	Percentuale di test di unità eseguiti - MPC21	25
3.3.2.1.22	Percentuale di test di integrazione eseguiti - MPC22	25
3.3.2.1.23	Percentuale di test di sistema eseguiti - MPC23	25
3.3.2.1.24	Percentuale di test di validazione eseguiti - MPC24	25
3.3.2.1.25	Percentuale dei test superati - MPC25	25

3.3.2.1.26	Percentuale di rami decisionali percorsi - MPC26	25
3.3.2.1.27	Numero di funzioni chiamate nei test - MPC27	26
3.3.2.1.28	Numero di istruzioni nei test - MPC28	26
3.3.2.2	Qualità di prodotto	26
3.3.2.2.1	Indice Gulpease - MPDD1	26
3.3.2.2.2	Completezza dell'implementazione funzionale - MPDS1	26
3.3.2.2.3	Percentuale di risultati concordi alle attese - MPDS2	26
3.3.2.2.4	Percentuale di operazioni illegali non bloccate - MPDS3	27
3.3.2.2.5	Percentuale failure su test-case - MPDS4	27
3.3.2.2.6	Numero di failure evitati - MPDS5	27
3.3.2.2.7	Percentuale delle funzionalità comprese - MPDS6	27
3.3.2.2.8	Percentuale di funzionalità conformi alle aspettative - MPDS7	27
3.3.2.2.9	Tempo medio di risposta - MPDS8	28
3.3.2.2.10	Percentuale di failure con cause individuate - MPDS9	28
3.3.2.2.11	percentuale di failure introdotte con modifiche - MPDS10	28
3.3.3	Procedure	28
3.3.3.1	Calcolo dell'indice di Gulpease	28
3.3.3.2	Controllo ortografico	28
3.3.3.3	Resoconto stato metriche	29
3.3.4	Strumenti	29
3.3.4.1	Script per il calcolo dell'indice di Gulpease	29
3.3.4.2	Controllo ortografico	29
3.3.4.3	Requisiti obbligatori soddisfatti	29
3.3.4.4	Requisiti accettati soddisfatti	29
3.3.4.5	Requisiti non accettati soddisfatti	29
3.3.4.6	Requisiti obbligatori soddisfatti	29
3.3.4.7	Structural Fan-In	29
3.3.4.8	Structural Fan-Out	29
3.3.4.9	Metodi per classe	30
3.3.4.10	Parametri per metodo	30
3.3.4.11	Componenti integrate	30
3.3.4.12	Test di unità eseguiti	30
3.3.4.13	Test di integrazione eseguiti	30
3.3.4.14	Test di sistema eseguiti	30
3.3.4.15	Test di validazione eseguiti	30
3.3.4.16	Test superati	30
3.3.4.17	Completezza implementazione funzionale	30
3.3.4.18	Densità di failure	30
4	Processi organizzativi	31
4.1	Gestione	31
4.1.1	Scopo	31
4.1.2	Aspettative	31
4.1.3	Descrizione	31
4.1.4	Ruoli di progetto	31
4.1.4.1	Responsabile	31
4.1.4.2	Amministratore	31
4.1.4.3	Analista	32
4.1.4.4	Progettista	32
4.1.4.5	Programmatore	32
4.1.4.6	Verificatore	32
4.1.5	Comunicazioni	32
4.1.5.1	Interne	32
4.1.5.2	Esterne	32
4.1.6	Incontri	33
4.1.6.1	Interni	33

4.1.6.2	Esterni	33
4.1.7	Strumenti di coordinamento	33
4.1.7.1	<i>Ticketing_g</i>	33
4.1.8	Strumenti di versionamento	33
4.1.8.1	<i>Repository_g</i>	33
4.1.8.2	Struttura del <i>repository_g</i> Docs	33
4.1.8.3	Commit	34
4.1.9	Rischi	34
4.1.10	Strumenti	34
4.1.10.1	Telegram	34
4.1.10.2	<i>Google Hangouts_g</i>	34
4.1.10.3	<i>Git_g</i>	34
4.1.10.4	<i>GitHub_g</i>	34
4.1.10.5	<i>GitHub_g</i> desktop	34
4.1.10.6	<i>Asana_g</i>	35
4.1.10.7	GanttProject	35

Elenco delle figure

1	Astah	12
2	Flow chart dell'approvazione di un documento	14
3	Texmaker	20
4	Github	35
5	Asana	35
6	GanttProject	36

1 Introduzione

1.1 Scopo del documento

Questo documento specifica e definisce le norme da rispettare all'interno del gruppo Co.Code durante lo svolgimento del *progetto_g* AtAVi.

Ogni membro del team è tenuto a visionare il documento e a rispettare le norme in esso contenute. Tali norme permettono di ottenere uniformità nei documenti sviluppati, migliorare l'efficienza del lavoro svolto e ridurre il numero di errori.

In particolare si tratteranno:

- le interazioni tra i membri del team;
- le interazioni del team con componenti esterne;
- le modalità di stesura dei documenti;
- la gestione del *repository_g*;
- le modalità di lavoro durante le varie fasi del *progetto_g*;
- l'ambiente di lavoro utilizzato.

In caso di modifiche o aggiunte a questo documento è necessario avvisare tutti i membri del gruppo.

1.2 Scopo del *prodotto_g*

Si vuole creare un'applicazione web che permetta ad un ospite, in visita all'ufficio di Zero12, di interrogare un assistente virtuale per annunciare la propria presenza, avvisare l'interessato del suo arrivo sul sistema di comunicazione aziendale (*Slack_g*) e nel frattempo essere intrattenuto con varie attività.

1.3 Glossario

Allo scopo di evitare ogni ambiguità nel linguaggio e rendere più semplice e chiara la comprensione dei documenti, viene allegato il "*Glossario v1.0.0*". Le parole in esso contenute sono scritte in corsivo e marcate con una 'g' a pedice (p.es. *Parola_g*).

1.4 Riferimenti

1.4.1 Riferimenti Normativi

- *Capitolato_g* d'appalto C2 - AtAVi: Accoglienza tramite Assistente Virtuale
<http://www.math.unipd.it/~tullio/IS-1/2016/Progetto/C2.pdf>;
- rappresentazione date
https://en.wikipedia.org/wiki/ISO_8601;
- composizione processo di sviluppo
https://en.wikipedia.org/wiki/ISO/IEC_12207.

1.4.2 Riferimenti Informativi

- *Git*_g
<https://git-scm.com/documentation>
- Slide del corso di ingegneria del *software*_g
 - <http://www.math.unipd.it/~tullio/IS-1/2016/Dispense/L02.pdf>
 - <http://www.math.unipd.it/~tullio/IS-1/2016/Dispense/L04.pdf>
 - <http://www.math.unipd.it/~tullio/IS-1/2016/Dispense/L05.pdf>
 - <http://www.math.unipd.it/~tullio/IS-1/2016/Dispense/L08.pdf>
 - <http://www.math.unipd.it/~tullio/IS-1/2016/Dispense/L12.pdf>

2 Processi primari

2.1 Sviluppo

2.1.1 Scopo

Include le attività e i compiti svolti per creare il *prodotto_g*.

2.1.2 Aspettative

Le aspettative della corretta implementazione del processo sono:

- realizzare un *prodotto_g* finale conforme alle richieste del *proponente_g* e che soddisfi le attività di *validazione_g* e *verifica_g*;
- fissare gli obiettivi di sviluppo;
- fissare i vincoli tecnologici.

2.1.3 Descrizione

In accordo con lo standard [ISO/IEC 12207], il processo di sviluppo è composto dalle attività di:

- analisi dei requisiti;
- progettazione;
- codifica;
- validazione.

2.1.4 Analisi dei requisiti

2.1.4.1 Scopo dell'attività

Individuare i requisiti del *progetto_g* dalle specifiche del *capitolato_g* e tramite incontri con il proponente. Tale attività produrrà un documento redatto dagli analisti, i quali avranno cura di elencare i *casi d'uso_g* e i requisiti. Tale documento permette di capire le scelte di progettazione effettuate.

2.1.4.2 Aspettative dell'attività

L'attività fissa come scopo la creazione di un documento che elencherà e rappresenterà i requisiti richiesti dal *proponente_g*.

2.1.4.3 Descrizione dell'attività

Tutti i requisiti analizzati, utilizzando le specifiche del *capitolato_g* e consultando i proponenti negli incontri effettuati, vanno specificati nell'“*Analisi dei Requisiti v1.0.0*”. Per analizzare e trovare i requisiti si utilizza la tecnica dei *casi d'uso_g*. Il tracciamento dei requisiti avviene tramite l'applicativo PragmaDB.

2.1.4.4 Studio di fattibilità

Il *Responsabile* di *progetto_g* deve organizzare delle riunioni preventive, per permettere lo scambio di opinioni tra i membri del gruppo sui capitoli proposti. Il documento *prodotto_g* da queste riunioni è lo “*Studio di Fattibilità v1.0.0*”, il quale viene realizzato dagli *Analisti*. Essi devono descrivere i seguenti punti:

- **Dominio tecnologico e applicativo:** si dà una valutazione prendendo in considerazione la conoscenza attuale delle tecnologie richieste dal *capitolato_g* in analisi da parte dei membri del gruppo;
- **Interesse strategico:** si valuta l'interesse strategico del gruppo di *progetto_g* in relazione al *capitolato_g* in analisi;
- **Individuazione dei rischi:** si analizzano i possibili rischi in cui si può incorrere nel *capitolato_g* in analisi.

2.1.4.5 Casi d'uso_g

Ogni caso d'uso è così composto:

- **Codice identificativo:** codice univoco del caso d'uso in esame;
- **Titolo:** indica il titolo del caso d'uso;
- **Diagramma *UML_g*:** rappresenta graficamente il caso d'uso;
- **Attori primari:** indica gli attori primari coinvolti;
- **Descrizione:** chiara, precisa e concisa descrizione del caso d'uso;
- **Precondizione:** indica la situazione che deve essere vera prima dell'esecuzione del caso d'uso;
- **Postcondizione** indica la situazione che deve essere vera dopo l'esecuzione del caso d'uso;
- **Scenario principale:** descrizione composta dal flusso dei *casi d'uso_g* figli;
- **Scenari alternativi:** descrizione composta dai *casi d'uso_g* che non appartengono al flusso principale di esecuzione.

2.1.4.6 Codice identificativo dei *casi d'uso_g*

Ogni caso d'uso ha un proprio codice identificativo che rispetta il seguente formalismo:

$$UC\{\text{Codice}\}$$

dove:

- **Codice:** indica il codice identificativo del requisito, è univoco e deve essere identificato in forma gerarchica.

2.1.4.7 Requisiti

Ogni requisito è così composto:

- **Codice identificativo:** codice univoco del requisito;
- **Descrizione:** una breve descrizione, deve essere meno ambigua possibile;
- **Fonti:** identifica la fonte dalla quale è stato identificato il requisito.

2.1.4.8 Codice identificativo dei requisiti

Ogni requisito individuato avrà un codice identificativo univoco così formato:

$$R\{\text{Tipo}\}\{\text{Importanza}\}\{\text{Codice}\}$$

dove:

- **Tipo:** può assumere uno di questi valori:
 - **F:** indica un requisito funzionale;
 - **Q:** indica un requisito di qualità;
 - **P:** indica un requisito prestazionale;
 - **V:** indica un requisito di vincolo.
- **Importanza:** può assumere uno di questi valori:
 - **O:** indica un requisito obbligatorio;
 - **D:** indica un requisito desiderabile;
 - **F:** indica un requisito facoltativo.
- **Codice:** indica il codice identificativo del requisito, è univoco e deve essere identificato in forma gerarchica.

2.1.4.9 UML_g

Viene utilizzata la versione corrente alla stesura del documento, ovvero la 2.5.

2.1.5 Progettazione

2.1.5.1 Scopo dell'attività

L'attività di progettazione definisce le linee essenziali della struttura del *prodotto_g software_g* in funzione dei requisiti individuati dall'analisi. L'obiettivo del processo consiste nella stesura dei documenti: "*Specifica Tecnica*" e "*Definizione di Prodotto_g*".

2.1.5.2 Aspettative dell'attività

Il processo porta alla formazione dei documenti sopracitati, i quali garantiscono affidabilità e coerenza.

2.1.5.3 Descrizione dell'attività

La progettazione deve rispettare tutti i vincoli e i requisiti concordati tra i componenti del gruppo e i proponenti. I documenti derivati da questa attività sono:

- **Specifica tecnica:** descrive la progettazione ad alto livello relativa all'architettura dell'applicazione e dei singoli componenti. Il documento specifica i diagrammi UML_g ed i design pattern utilizzati per realizzare l'architettura definendo inoltre i test necessari alla *verifica_g*;
- **Definizione di *prodotto_g*:** descrive in dettaglio la progettazione di *sistema_g*, integrando quanto scritto nella Specifica Tecnica. Il documento specifica i diagrammi UML_g e le definizioni delle classi definendo inoltre i test necessari alla *verifica_g*.

2.1.5.4 Specifica tecnica

- **Diagrammi UML_g :**
 - diagrammi delle classi;
 - diagrammi dei $package_g$;
 - diagrammi di attività;
 - diagrammi di sequenza.
- **$Design\ pattern_g$:** devono essere descritti i $design\ pattern_g$ utilizzati per realizzare l'architettura. Ogni design pattern deve essere accompagnato da una descrizione ed un diagramma, che ne esponga il significato e la struttura;
- **Tracciamento delle componenti:**
- **Test di integrazione:** devono essere definite delle classi di $verifica_g$, utili a verificare che ogni componente del $sistema_g$ funzioni nella maniera appropriata.

2.1.5.5 Definizione di $prodotto_g$

- **Diagrammi UML_g :**
 - diagrammi delle classi;
 - diagrammi di attività;
 - diagrammi di sequenza.
- **Definizioni delle classi:** ogni classe progettata deve essere descritta in modo da spiegarne lo scopo e definirne le funzionalità ad essa associate.
- **Tracciamento delle classi:** ogni requisito deve essere tracciato, in modo da poter risalire alle classi ad esso associate.
- **Test di unità:** devono essere definiti dei test di unità utili a verificare che le componenti del $sistema_g$ funzionino nel modo previsto.

2.1.6 Codifica

2.1.6.1 Scopo dell'attività

Lo scopo dell'attività è l'implementazione del $prodotto_g$, concretizzando la soluzione tramite la codifica.

2.1.6.2 Aspettative dell'attività

L'aspettativa dell'attività è un $prodotto_g$ corretto, ovvero stabile, affidabile, funzionale e che soddisfi i requisiti.

2.1.6.3 Descrizione dell'attività

L'attività deve rispettare i compiti e gli strumenti espressi nel “*Piano di Progetto v1.0.0*”.

2.1.6.4 Stile

Le norme di stile saranno specificate in versioni successive di questo documento.

2.1.6.5 Versionamento

Lo stile di rappresentazione della versione del codice verrà trattato e descritto in versioni successive di questo documento.

2.1.6.6 Ricorsione_g

La *ricorsione_g* va evitata. Se non risulta accettabile convertirla in *iterazione_g*, bisogna fornirne la prova di terminazione e l'analisi del costo in termini di spazio.

2.1.7 Strumenti

2.1.7.1 PragmaDB

PragmaDB è uno strumento *open source_g* di tracciamento dei requisiti. Verrà quindi utilizzato per semplificare e automatizzare il più possibile l'attività di analisi dei requisiti. In particolare, una volta inseriti *casi d'uso_g*, attori, requisiti e relative fonti, PragmaDB genera:

- il codice $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ relativo a *casi d'uso_g* e requisiti in forma tabellare;
- i diagrammi *UML_g* associati ai *casi d'uso_g*.

Essendo *open source_g*, questo strumento è stato adattato dal team Co.Code in base alle proprie necessità.

2.1.7.2 Astah_g

Astah_g è uno strumento di modellazione *UML_g*. Qualora i diagrammi *UML_g* generati da PragmaDB non siano soddisfacenti, si ricorrerà all'utilizzo di *Astah_g*. Viene utilizzata la versione 7.0 o superiori.

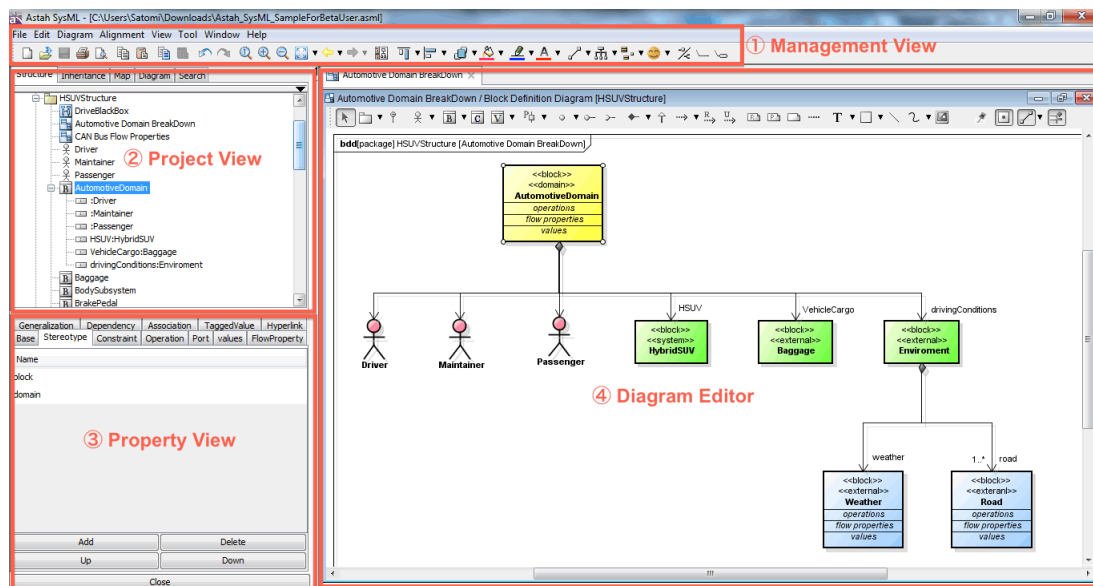


Figura 1: Astah

3 Processi di supporto

3.1 Documentazione

3.1.1 Scopo

Lo scopo di questo processo consiste nell'illustrazione di come deve essere redatta e mantenuta la documentazione, durante il *ciclo di vita_g* del *software_g*.

3.1.2 Aspettative

Le aspettative della corretta implementazione di tale processo sono:

- una chiara visione della documentazione prodotta durante il *ciclo di vita_g* del *software_g*;
- una serie di norme per la stesura di documenti coerenti e validi;
- una documentazione formale e coerente.

3.1.3 Descrizione

In questo documento devono essere redatte tutte le norme e le convenzioni adottate dal gruppo, in modo da produrre una documentazione valida e coerente.

3.1.4 Procedure

Per la stesura della documentazione si è utilizzato il linguaggio \LaTeX , si veda la [sezione strumenti 3.1.11](#).

3.1.4.1 Approvazione dei documenti

La formalizzazione di un documento segue la seguente procedura:

1. il documento viene redatto da coloro che sono incaricati della sua stesura ed eventuale correzione di errori;
2. per ogni significativa modifica del documento, i *Verificatori* avranno il compito di controllare la presenza di errori o imprecisioni;
3. se i *Verificatori* riscontrano degli errori, dovranno notificarlo ai redattori del documento tramite una specifica *issue_g*, tornando così al punto 1, altrimenti, se completo, il documento viene consegnato al *Responsabile*;
4. il *Responsabile* di *progetto_g* decide se approvare, e quindi formalizzare il documento, oppure se rifiutarlo comunicando la motivazione e le modifiche da apportare, tornando così al punto 1.

3.1.5 Template

Per garantire omogeneità tra i documenti è stato creato un *template_g* \LaTeX , dove sono state definite tutte le regole di formattazione da applicare al documento. Questo permette a tutti i componenti del gruppo di concentrarsi solo nella stesura del contenuto, senza doversi preoccupare dell'aspetto.

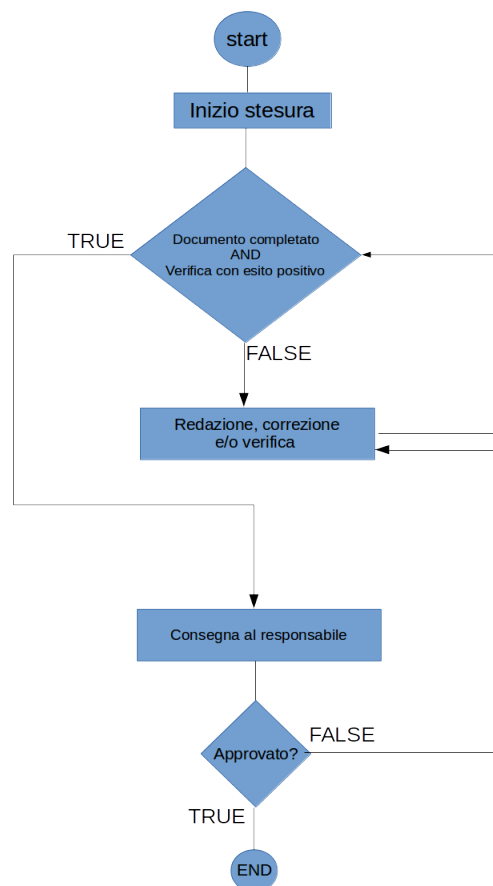


Figura 2: Flow chart dell'approvazione di un documento

3.1.6 Struttura dei documenti

3.1.6.1 Frontespizio

La prima pagina di ogni documento dovrà contenere:

- logo del gruppo;
- nome del *progetto*_g;
- nome del documento e la relativa versione;
- sommario;
- data di redazione;
- nome e cognome dei redattori del documento;
- nome e cognome dei verificatori del documento;
- nome e cognome del responsabile per l'approvazione del documento;
- uso del documento (interno o esterno);
- lista di distribuzione del documento.

3.1.6.2 Diario delle modifiche

La seconda pagina dovrà contenere il diario delle modifiche di quel determinato documento. Il diario è costituito di una tabella ordinata in modo decrescente seconda la data di modifica e il numero di versione.

Gli attributi della tabella rappresentano:

- numero di versione;
- breve riepilogo delle modifiche apportate;
- autore delle modifiche;
- ruolo ricoperto dall'autore all'interno del *progetto_g*;
- data di modifica.

3.1.6.3 Indice

In ogni documento è presente un indice delle sezioni, utile a fornire una visione macroscopica della struttura del documento. Sono previsti, se necessari, gli indici relativi alle tabelle e alle figure presenti nel documento in questo ordine.

3.1.6.4 Intestazione e piè di pagina

L'intestazione delle pagine di ogni documento deve contenere:

- numero e titolo della sezione;
- nome del *progetto_g*.

Il piè di pagina contiene invece:

- nome del documento con la relativa versione;
- nome del gruppo;
- pagina X di Y, dove X è la pagina corrente e Y è il numero di pagine totali del documento.

3.1.7 Versionamento

Ciascun documento che verrà redatto dovrà essere versionato, per consentire un tracciamento chiaro della sua storia e delle sue modifiche.

Verrà applicato il seguente formalismo:

vX.Y.Z

dove:

- **X:**
 - inizia da 0;
 - viene incrementato quando il *Responsabile di progetto_g* approva il documento.
- **Y:**
 - inizia da 0;
 - viene incrementato dal *Verificatore* ad ogni *verifica_g*;
 - quando viene modificato X, viene riportato a 0.

- **Z:**
 - inizia da 0;
 - viene incrementato dal Redattore del documento dopo ogni modifica;
 - quando viene modificato Y, viene riportato a 0.

3.1.8 Norme tipografiche

In questa sezione vengono definite le norme ortografiche e tipografiche da rispettare nella stesura di ogni documento.

3.1.8.1 Stile del testo

- **Grassetto:** viene utilizzato per:
 - titoli;
 - elementi di un elenco puntato che riassumono il contenuto del relativo paragrafo.
- **Corsivo:** viene utilizzato per:
 - citazioni;
 - abbreviazioni;
 - parole inserite nel glossario;
 - riferimenti ad altri documenti;
 - nomi di società o aziende;
 - ruoli dei membri del gruppo.
- **Maiuscolo:** le parole scritte interamente in maiuscolo dovranno riferirsi soltanto ad acronimi.
- **Monospace:** le porzioni di testo scritte in monospace definiscono:
 - frammenti di codice;
 - comandi;
 - URL.
- **Glossario:** le parole che hanno un riferimento nel glossario sono in corsivo e hanno una 'g' a pedice.

3.1.8.2 Elenchi puntati

Tutti gli elenchi puntati sono caratterizzati graficamente da un pallino nel primo livello, da una trattino nel secondo e da un asterisco nel terzo (automatizzato grazie al *template_g* L^AT_EX creato). Ogni elemento deve terminare con il punto e virgola, a meno che non sia l'ultimo dell'elenco, in questo caso la frase va terminata con il punto. Ogni punto inizia con la minuscola, tranne nel caso in cui necessiti di una spiegazione: allora si utilizzerà la maiuscola.

3.1.8.3 Formati comuni

- **Date:**

AAAA - MM - GG

dove:

- AAAA: rappresenta l'anno utilizzando 4 cifre;
- MM: rappresenta il mese utilizzando 2 cifre;
- GG: rappresenta il giorno utilizzando 2 cifre.

- **Orari:**

HH:MM

dove:

- HH: rappresenta l'ora e può assumere valori da 0 a 23;
- MM: rappresenta i minuti e può assumere valori da 0 a 59.

- **Nomi ricorrenti:**

- **Ruoli di *progetto*_g**: ogni nome di un ruolo di *progetto*_g deve essere scritto con la lettera iniziale maiuscola e con lo stile corsivo. Questo viene automatizzato utilizzando il comando "`\CodiceRuolo`";
- **Nomi propri**: ogni nome deve essere espresso nella forma "Nome Cognome";
- **Nomi dei documenti**: ogni nome di documento viene scritto con lo stile corsivo, con l'iniziale di ogni parola maiuscola e con la versione corrente. Questo viene automatizzato richiamando il comando "`\SiglaDocumentodoc`".

3.1.8.4 Sigle

E' previsto l'utilizzo di queste sigle:

- **AdR**: per "*Analisi dei Requisiti v1.0.0*";
- **PdP**: per "*Piano di Progetto v1.0.0*";
- **NdP**: per "*Norme di Progetto v1.0.0*";
- **SdF**: per "*Studio di Fattibilità v1.0.0*";
- **PdQ**: per "*Piano di Qualifica v1.0.0*";
- **ST**: per "*Specifica Tecnica v1.0.0*";
- **Gl**: per "*Glossario v1.0.0*";
- **DP**: per "*Definizione di Prodotto v1.0.0*".

3.1.9 Elementi grafici

3.1.9.1 Tabelle

Le tabelle devono essere accompagnate da una didascalia e da un numero incrementale per garantirne la tracciabilità.

3.1.9.2 Immagini

Ogni immagine deve essere centrata orizzontalmente. Inoltre deve essere nettamente separata dai paragrafi che la seguono e la precedono, in modo da definire un netto distacco tra testo e grafica e migliorare conseguentemente la leggibilità. Essa dev'essere accompagnata da una didascalia

analoga a quella descritta per le tabelle. Tutti i diagrammi *UML_g* vengono inseriti nel documento sotto forma di immagine.

3.1.10 Classificazione dei documenti

3.1.10.1 Documenti informali

Tutti i documenti sono da ritenersi informali fino all'approvazione da parte del *Responsabile* di *progetto_g*, ed in quanto tali sono da considerarsi esclusivamente ad uso interno.

3.1.10.2 Documenti formali

Un documento viene definito formale quando viene validato dal *Responsabile* di *progetto_g*. Solo i documenti formali possono essere distribuiti all'esterno del gruppo. Per arrivare a tale stato il documento deve aver passato la *verifica_g* e la *validazione_g*.

3.1.10.3 Glossario

Il glossario nasce dall'esigenza di chiarire il significato di parole che possono risultare ambigue all'interno di determinati contesti. Saranno quindi presenti parole che:

- trattano argomenti tecnici;
- possono creare delle ambiguità sul significato;
- rappresentano delle sigle.

La struttura deve avere queste caratteristiche:

- le parole devono essere in ordine alfabetico;
- ogni termine deve essere seguito da una spiegazione chiara e concisa, che non generi alcun tipo di ambiguità.

3.1.10.4 Verbali

Questo documento ha lo scopo di riassumere in modo formale le discussioni effettuate e le decisioni prese durante le riunioni. I verbali, come le riunioni, sono classificati in: interni ed esterni. In particolare i verbali esterni, essendo documenti ufficiali, devono essere redatti dal *Responsabile* di Progetto. Ogni verbale dovrà essere denominato nel seguente modo:

Verbale_TipoVerbale_DataVerbale

dove:

- **TipoVerbale:** identifica se il verbale è riferito ad una riunione interna (I) o esterna (E);
- **DataVerbale:** identifica la data nella quale si è svolta la riunione relativa al verbale.

Nella parte introduttiva vengono specificate le seguenti informazioni:

- luogo di incontro;
- data di incontro;
- orario di inizio;
- orario di fine;
- durata dell'incontro;

- oggetto dell'incontro;
- partecipanti;
- segretario;
- segnalazioni varie.

Tutte le decisioni prese durante la riunione vengono identificate univocamente utilizzando questo formato:

DIX.Y per i verbali interni

DEX.Y per i verbali esterni

dove:

- **X**: rappresenta il numero di verbale redatto in ordine cronologico (inizia da 1);
- **Y**: rappresenta il numero della decisione all'interno di un singolo verbale (inizia da 1).

Inoltre vengono tracciate le decisioni in sospeso che verranno chiarite in verbali successivi. Il formato identificativo è lo stesso delle decisioni definitive, dove nel codice la D viene sostituita dalla S.

3.1.11 Strumenti

3.1.11.1 \LaTeX

La stesura dei documenti deve essere effettuata utilizzando il linguaggio di *markup*_g \LaTeX . Le motivazioni di questa scelta sono dovute alle possibilità che \LaTeX offre:

- creazione di documenti formali in modo rapido ed efficiente;
- possibilità di separare contenuto e formattazione, definendo l'aspetto delle pagine in un file *template*_g separato e condiviso da tutti i documenti;
- creazione e gestione automatica dell'indice del documento.

Inoltre è stato reso disponibile uno script *PHP*_g che per ogni documento \LaTeX marca tutte le parole presenti nel “*Glossario v1.0.0*” secondo le regole decise nelle “*Norme di Progetto v1.0.0*”.

3.1.11.2 *Texmaker*_g

Per la redazione del codice \LaTeX viene utilizzato l'editor *Texmaker*_g. Questo strumento oltre ad integrare un compilatore e visualizzatore *PDF*_g, fornisce suggerimenti per il completamento dei comandi \LaTeX .

3.1.11.3 Excel

Per la creazione di grafici (istogrammi, diagrammi a torta, ecc.) viene utilizzato Excel di Microsoft Office, nella versione 2013 o successive.

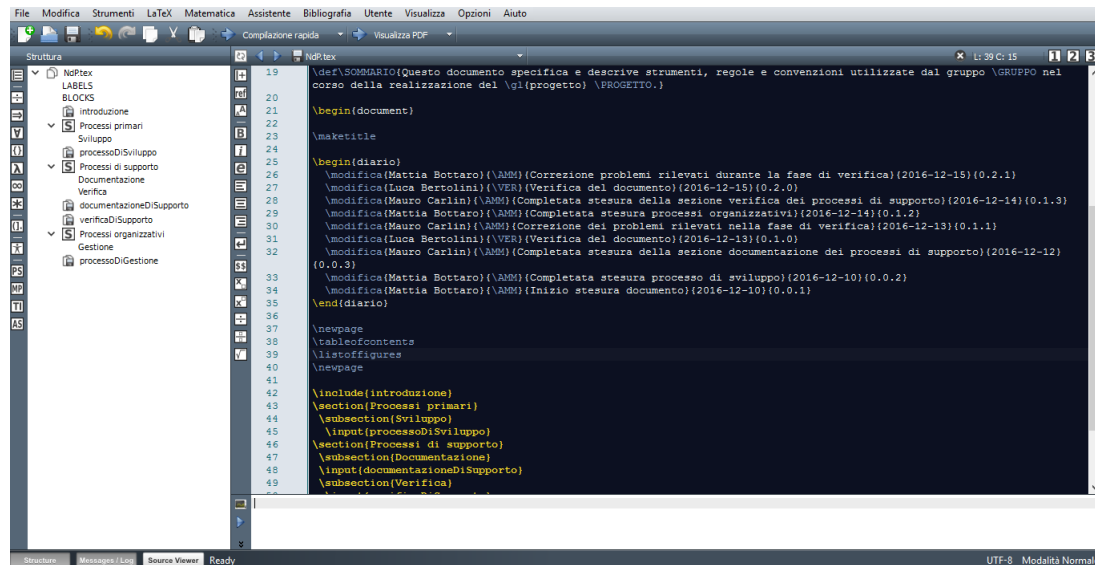


Figura 3: Texmaker

3.2 Verifica_g

3.2.1 Scopo del processo

Si occupa di accertare che lo svolgimento del processo in esame non introduca errori nel *prodotto_g*.

3.2.2 Aspettative del processo

Una corretta implementazione di tale processo permette di individuare:

- una procedura di *verifica_g*;
- i criteri per la *verifica_g* del *prodotto_g*.

3.2.3 Attività

3.2.3.1 Analisi statica

E' una tecnica di analisi del codice sorgente e della documentazione associata, prevalentemente usata quando il *sistema_g* non è ancora disponibile e durante tutto l'arco del suo sviluppo. Non richiede l'esecuzione del *prodotto_g software_g* in alcuna sua parte. Può essere applicata tramite una delle seguenti strategie:

- **Walkthrough:** si legge l'intero documento (o codice) in cerca di tutte le possibili anomalie. E' una tecnica onerosa che richiede l'impegno di più persone e per questo deve essere utilizzata solo durante la prima parte del *progetto_g*, dove non tutti i membri hanno piena padronanza e conoscenza delle “*Norme di Progetto v1.0.0*” e del “*Piano di Qualifica v1.0.0*”;
- **Inspection:** questa tecnica dev'essere applicata quando si ha idea della problematica che si sta cercando; consiste in una lettura mirata del documento (o del codice), sulla base di una lista degli errori precedentemente stilata.

3.2.3.2 Analisi dinamica

L'attività di analisi dinamica è una tecnica di *verifica_g* applicabile solamente al *software_g*. Tale tecnica può essere utilizzata per analizzare l'intero *software_g* o una porzione limitata dello stesso. L'attività consiste nell'esecuzione di test automatici realizzati dal team. Le verifiche devono essere effettuate su un insieme finito di casi, con valori di ingresso, uno stato iniziale e un esito decidibile. Tutti i test producono risultati automatici che inviano notifiche sulla tipologia di problema individuato. Ogni test è ripetibile, ossia applicabile durante l'intero *ciclo di vita_g* del *software_g*.

3.3 Qualità

3.3.1 Notazione

3.3.1.1 Metriche

Per garantire la qualità del lavoro del team gli *Amministratori* hanno definito delle metriche, riportandole nel “*Piano di Qualifica v1.0.0*”, che devono rispettare la seguente notazione:

$$M\{X\}\{Y\}\{Z\}$$

dove:

- **X** indica se la metrica si riferisce a prodotti o processi e può assumere i valori:
 - **PC** per indicare i processi;
 - **PD** per indicare i prodotti.
- **Y** presente solo se la metrica è riferita ai prodotti, indica se il termine *prodotto_g* si riferisce a documenti o al software e può assumere i seguenti valori:
 - **D** per indicare i documenti;
 - **S** per indicare il software;
- **Z** indica il codice univoco della metrica (numero intero incrementale a partire da 1).

3.3.1.2 Obiettivi

Per garantire la qualità del lavoro del team, gli *Amministratori* hanno definito degli obiettivi di qualità, riportandoli nel “*Piano di Qualifica v1.0.0*”, che devono rispettare la seguente notazione:

$$O\{X\}\{Y\}\{Z\}$$

dove:

- **X** indica se l'obiettivo si riferisce a prodotti o processi e può assumere i valori:
 - **PC** per indicare i processi;
 - **PD** per indicare i prodotti.
- **Y** presente solo se l'obiettivo è riferito ai prodotti, indica se il termine *prodotto_g* si riferisce a documenti o al software e può assumere i seguenti valori:
 - **D** per indicare i documenti;
 - **S** per indicare il software;
- **Z** indica il codice univoco dell'obiettivo (numero intero incrementale a partire da 1).

3.3.2 Definizione metriche

Di seguito sono definite le metriche utilizzate nel documento “*Piano di Qualifica v2.0.0*”. Ad ogni metrica è stata assegnato un codice identificativo per facilitare il tracciamento.

3.3.2.1 Qualità di processo

3.3.2.1.1 Percentuale di accessi avvenuti correttamente a PragmaDB - MPC1

Indica il numero di accessi avvenuti correttamente espresso in percentuale.

$$\text{Percentuale accessi} = \frac{\text{Accessi avvenuti}}{\text{Richieste d'accesso}} * 100$$

3.3.2.1.2 Schedule Variance - MPC2

Indica se si è in linea, in anticipo o in ritardo rispetto ai tempi pianificati. Il risultato può essere:

- positivo: si è in anticipo rispetto alla pianificazione;
- negativo: si è in ritardo rispetto alla pianificazione;
- nullo: si è in linea rispetto alla pianificazione;

$$\text{Schedule Variance} = \text{TP} - \text{TR}$$

- **TP** è il tempo pianificato per terminare un attività;
- **TR** è il tempo reale che è stato impiegato.

3.3.2.1.3 Cost Variance in percentuale - MPC3

Indica se alla data corrente i costi corrispondono alla pianificazione, espressi in percentuale. Il risultato può essere:

- positivo: i costi sono minori di quelli pianificati;
- negativo: i costi sono maggiori di quelli pianificati;
- nullo: i costi sono uguali a quelli pianificati.

$$\text{Cost Variance} = \frac{CP - CR}{CP} * 100$$

- **CP** sono i costi pianificati per la data corrente;
- **CR** sono i costi reali sostenuti.

3.3.2.1.4 Indice dei rischi non preventivati - MPC4

É un indice che viene incrementato ogniqualvolta si manifesta un rischio non individuato nell'attività di analisi dei rischi.

3.3.2.1.5 Numero di requisiti obbligatori soddisfatti - MPC5

Indica la percentuale di requisiti obbligatori soddisfatti del prodotto.

$$Numerorequisiti = \frac{\text{requisiti obbligatori soddisfatti}}{\text{requisiti obbligatori totali}} * 100$$

3.3.2.1.6 Numero di requisiti desiderabili soddisfatti - MPC6

Indica la percentuale di requisiti desiderabili soddisfatti del prodotto.

$$\text{Numero requisiti} = \frac{\text{requisiti desiderabili soddisfatti}}{\text{requisiti desiderabili totali}} * 100$$

3.3.2.1.7 Numero di requisiti opzionali soddisfatti - MPC7

Indica la percentuale di requisiti opzionali soddisfatti del prodotto.

$$\text{Numero requisiti} = \frac{\text{requisiti opzionali soddisfatti}}{\text{requisiti opzionali totali}} * 100$$

3.3.2.1.8 SF-IN - MPC8

Indice numerico che incrementa nel momento in cui viene individuato un modulo che, durante la sua esecuzione, chiama il modulo in oggetto.

3.3.2.1.9 SF-OUT - MPC9

Indice numerico che incrementa nel momento in cui viene individuato un modulo utilizzato dal modulo in oggetto durante la sua esecuzione.

3.3.2.1.10 ???? - MPC10**3.3.2.1.11 Numero di metodi per classe - MPC11**

Indica il numero di metodi definiti in ogni classe.

3.3.2.1.12 Numero di parametri per metodo - MPC12

Indica il numero di parametri definiti in ogni metodo.

3.3.2.1.13 Indice di complessità ciclomatica - MPC13

Dato un grafo non fortemente connesso che rappresenta una sezione di codice del software, indica il massimo numero di cammini percorribili possibili tra loro indipendenti.

$$\text{Complessità ciclomatica} = E - N + 2P$$

- **N** è il numero di nodi del grafo, che corrisponde a gruppi indivisibili di istruzioni;
- **E** è il numero di archi del grafo, presenti qualora il gruppo di istruzioni corrispondenti al nodo sottostante possono essere eseguite immediatamente dopo le istruzioni del nodo sovrastante;

- **P** è il numero di componenti connesse, ossia il numero di sottografi in cui qualsiasi coppia di vertici è connessa da cammini e tale sottografo non è connesso a nessun vertice addizionale del supergrafo.

Se il grafo è fortemente connesso, ossia ogni punto di uscita è a sua volta connesso col punto di ingresso, la complessità ciclomatica viene calcolata con la seguente formula:

$$\text{Complessità ciclomatica} = E - N + P$$

3.3.2.1.14 Numero di livelli di annidamento - MPC14

Indica il numero di procedure e funzioni annidate, ovvero richiamate all'interno di altre procedure o funzioni.

3.3.2.1.15 Percentuale di linee di commento per linee di codice - MPC15

Indica la percentuale di linee di commento rispetto alle linee di codice.

$$\text{Percentuale linee di commento} = \frac{\text{Linee di commento}}{\text{Linee di codice}} * 100$$

3.3.2.1.16 Halstead Difficulty per funzione - MPC16

Indica il livello di complessità di una funzione.

$$\text{Halstead Difficulty} = \frac{UOP}{2} * \frac{OD}{UOD}$$

- **UOP** è il numero di operatori distinti;
- **OD** è il numero totale di operandi;
- **UOD** è il numero di operandi distinti.

3.3.2.1.17 Halstead Volume per funzione - MPC17

Indica la dimensione dell'implementazione di un algoritmo.

$$\text{Halstead Volume} = (OP + OD) * \log_2(UOP + UOD)$$

- **OP** è il numero totale di operatori;
- **UOP** è il numero di operatori distinti;
- **OD** è il numero totale di operandi;
- **UOD** è il numero di operandi distinti.

3.3.2.1.18 Halstead Effort per funzione - MPC18

Indica il costo necessario a scrivere il codice di una funzione.

$$\text{Halstead Effort} = \text{Halsted Difficulty} * \text{Halstead Volume}$$

3.3.2.1.19 Indice di manutenibilità - MPC19

Indica quanto sarà semplice mantenere il codice prodotto. I valori possibili vanno da 0 a 100 e sono così interpretati:

- un valore compreso tra 20 e 100 indica che il codice è facilmente manutenibile;
- un valore compreso tra 10 e 19 indica che il codice è moderatamente manutenibile;
- un valore compreso tra 0 e 9 indica che il codice è difficilmente manutenibile.

Manutenibilità = $171 - 5,2 * \ln(\text{Halstead Volume}) - 0.23 * (\text{Complessità ciclomatica}) - 16.2 * \ln(\text{Linee di codice})$

3.3.2.1.20 Percentuale di componenti integrate nel sistema - MPC20

Indica la percentuale di componenti attualmente implementate e correttamente integrate nel sistema.

$$\text{Componenti integrate} = \frac{\text{Numero componenti integrate}}{\text{Numero componenti totali progettate}} * 100$$

3.3.2.1.21 Percentuale di test di unità eseguiti - MPC21

Indica la percentuale di test di unità eseguiti.

$$\text{Test di unità eseguiti} = \frac{\text{Numero test di unità eseguiti}}{\text{Numero test di unità pianificati}} * 100$$

3.3.2.1.22 Percentuale di test di integrazione eseguiti - MPC22

Indica la percentuale di test di integrazione eseguiti.

$$\text{Test di integrazione eseguiti} = \frac{\text{Numero test di integrazione eseguiti}}{\text{Numero test di integrazione pianificati}} * 100$$

3.3.2.1.23 Percentuale di test di sistema eseguiti - MPC23

Indica la percentuale di test di sistema eseguiti.

$$\text{Test di sistema eseguiti} = \frac{\text{Numero test di sistema eseguiti}}{\text{Numero test di sistema pianificati}} * 100$$

3.3.2.1.24 Percentuale di test di validazione eseguiti - MPC24

Indica la percentuale di test di validazione eseguiti.

$$\text{Test di validazione eseguiti} = \frac{\text{Numero test di validazione eseguiti}}{\text{Numero test di validazione pianificati}} * 100$$

3.3.2.1.25 Percentuale dei test superati - MPC25

Indica la percentuale di test superati.

$$\text{Test superati} = \frac{\text{Numero test superati}}{\text{Numero test eseguiti}} * 100$$

3.3.2.1.26 Percentuale di rami decisionali percorsi - MPC26

Indica la percentuale di rami decisionali percorsi dai test di unità utilizzati.

$$\text{Rami decisionali percorsi} = \frac{\text{Numero rami decisionali percorsi}}{\text{Numero rami decisionali totali}} * 100$$

3.3.2.1.27 Numero di funzioni chiamate nei test - MPC27

Indica il numero di funzioni chiamate nel test di unità utilizzato.

3.3.2.1.28 Numero di istruzioni nei test - MPC28

Indica il numero di istruzioni eseguite nel test di unità utilizzato.

3.3.2.2 Qualità di prodotto**3.3.2.2.1 Indice Gulpease - MPDD1**

Permette di calcolare il livello di leggibilità e comprensibilità del documento. I risultati sono compresi tra 0 e 100, dove il valore "100" indica la leggibilità più alta e "0" la leggibilità più bassa. In generale risulta che testi con un indice:

- inferiore a 80 sono difficili da leggere per chi ha la licenza elementare;
- inferiore a 60 sono difficili da leggere per chi ha la licenza media;
- inferiore a 40 sono difficili da leggere per chi ha un diploma superiore.

$$\text{Indice Gulpease} = 89 + \frac{300 * A + 10 * B}{C}$$

- **A** è il numero totale di frasi;
- **B** è il numero totale di lettere
- **C** è il numero totale di parole;

3.3.2.2.2 Completezza dell'implementazione funzionale - MPDS1

Permette di calcolare il livello di leggibilità e comprensibilità del documento.

$$C = \left(1 - \frac{FM}{FI}\right) \cdot 100$$

- **FM** è il numero di funzionalità mancanti nell'implementazione;
- **FI** è il numero di funzionalità individuate nell'attività di analisi;

3.3.2.2.3 Percentuale di risultati concordi alle attese - MPDS2

Calcola quanti risultati sono concordi alle attese.

$$RC = \left(1 - \frac{N_{RD}}{N_{TE}}\right) \cdot 100$$

- **RD** è il numero di test che producono risultati discordanti rispetto alle attese;
- **TE** è il numero di test-case eseguiti;

3.3.2.2.4 Percentuale di operazioni illegali non bloccate - MPDS3

Calcola quante operazioni illegali non sono state bloccate.

$$I = \frac{N_{IE}}{N_{II}} \cdot 100$$

- **IE** è il numero di operazioni illegali effettuabili dai test;
- **II** è il numero di operazioni illegali individuate;

3.3.2.2.5 Percentuale failure su test-case - MPDS4

Calcola la percentuale di operazioni di testing che si sono concluse in failure.

$$F = \frac{N_{FR}}{N_{TE}} \cdot 100$$

- **FR** è il numero di failure rilevati durante l'attività di testing;
- **TE** è il numero di test-case eseguiti;

3.3.2.2.6 Numero di failure evitati - MPDS5

Calcola la percentuale di funzionalità in grado di gestire correttamente i fault che potrebbero verificarsi.

$$B = \frac{N_{FE}}{N_{ON}} \cdot 100$$

- **FE** è il numero di failure evitati durante i test effettuati;
- **ON** è il numero di test-case eseguiti che prevedono l'esecuzione di operazioni non corrette, causa di possibili failure;

3.3.2.2.7 Percentuale delle funzionalità comprese - MPDS6

Calcola la percentuale di operazioni comprese in modo immediato dall'utente, senza la consultazione del manuale.

$$C = \frac{N_{FC}}{N_{FO}} \cdot 100$$

- **FC** è il numero di funzionalità comprese in modo immediato dall'utente durante l'attività di testing del prodotto;
- **FO** è il numero di funzionalità offerte dal sistema;

3.3.2.2.8 Percentuale di funzionalità conformi alle aspettative - MPDS7

Calcola la percentuale di funzionalità offerte all'utente che rispettano le sue aspettative riguardo al comportamento del software.

$$C = (1 - \frac{N_{MFI}}{N_{MFO}}) \cdot 100$$

- **MFI** è il numero di messaggi e funzionalità che non rispettano le aspettative dell'utente;
- **MFO** è il numero di messaggi e funzionalità offerti dal sistema;

3.3.2.2.9 Tempo medio di risposta - MPDS8

Calcola il periodo temporale medio trascorso tra la richiesta al software di una determinata funzionalità e la risposta all'utente.

$$T_{RISP} = \frac{\sum_{i=1}^n T_i}{n}$$

- T_{RISP} espresso in *secondi*;
- T_i è il tempo intercorso fra la richiesta i di una funzionalità ed il completamento delle operazioni necessarie a restituire un risultato a tale richiesta;

3.3.2.2.10 Percentuale di failure con cause individuate - MPDS9

Calcola la percentuale di failure di cui sono state individuate le cause.

$$I = \frac{N_{FI}}{N_{FR}} \cdot 100$$

- **FI** è il numero di failure delle quali sono state individuate le cause;
- **FR** è il numero di failure rilevate;

3.3.2.2.11 percentuale di failure introdotte con modifiche - MPDS10

Calcola la percentuale di modifiche effettuate in risposta a failure che hanno portato all'introduzione di nuove failure in altre componenti del sistema.

$$I = \frac{N_{FRF}}{N_{FR}} \cdot 100$$

- **FRF** è il numero di failure risolte con l'introduzione di nuove failure;
- **FR** è il numero di failure risolte;

3.3.3 Procedure**3.3.3.1 Calcolo dell'indice di Gulpease**

Affinché un documento possa superare la fase di approvazione, è necessario che soddisfi il test di leggibilità con un indice Gulpease superiore a 40 punti. Per valutare questa metrica di qualità, è necessario seguire la seguente procedura:

- dirigersi con il terminale in GulScript;

- dare il comando `php gulpease.php`;
- visualizzare il risultato sul terminale.

3.3.3.2 Controllo ortografico

Per verificare la correttezza ortografica è necessario seguire la seguente procedura:

- aprire Texmaker;
- aprire il documento interessato nel formato `.tex`;
- dal menù a tendina "Modifica", selezionare la voce "verifica ortografia".

Per rendere ciò possibile, è necessario installare il pacchetto relativo dizionario italiano per Texmaker.

3.3.3.3 Resoconto stato metriche

Per visualizzare il resoconto corrente dello stato di ciò che le metriche indicano, è necessario seguire la seguente procedura:

- effettuare l'accesso in PragmaDB;
- selezionare la voce "Metriche".

3.3.4 Strumenti

3.3.4.1 Script per il calcolo dell'indice di Gulpease

In GulScript si trova lo script che calcola l'indice di Gulpease per ogni documento.

3.3.4.2 Controllo ortografico

Per il controllo ortografico dei documenti si farà utilizzo dello strumento integrato in Texmaker. Per poterlo utilizzare, è necessario disporre del pacchetto per il dizionario italiano.

3.3.4.3 Requisiti obbligatori soddisfatti

Lo strumento scelto per il calcolo del valore di questa metrica è PragmaDB, il quale permette di tracciare i requisiti ed associarli su use case e fonti.

3.3.4.4 Requisiti accettati soddisfatti

Lo strumento scelto per il calcolo del valore di questa metrica è PragmaDB, il quale permette di tracciare i requisiti ed associarli su use case e fonti.

3.3.4.5 Requisiti non accettati soddisfatti

Lo strumento scelto per il calcolo del valore di questa metrica è PragmaDB,

3.3.4.6 Requisiti obbligatori soddisfatti

Lo strumento scelto per il calcolo del valore di questa metrica è PragmaDB, il quale permette di tracciare i requisiti ed associarli su use case e fonti.

3.3.4.7 Structural Fan-In

Lo strumento scelto per il calcolo del valore di questa metrica è PragmaDB,

3.3.4.8 Structural Fan-Out

Lo strumento scelto per il calcolo del valore di questa metrica è PragmaDB,

3.3.4.9 Metodi per classe

Lo strumento scelto per il calcolo del valore di questa metrica è PragmaDB, il quale permette di tracciare le classi ed associarle ad altre classi correlate e requisiti.

3.3.4.10 Parametri per metodo

Lo strumento scelto per il calcolo del valore di questa metrica è PragmaDB,

3.3.4.11 Componenti integrate

Lo strumento scelto per il calcolo del valore di questa metrica è PragmaDB,

3.3.4.12 Test di unità eseguiti

Lo strumento scelto per il calcolo del valore di questa metrica è PragmaDB,

3.3.4.13 Test di integrazione eseguiti

Lo strumento scelto per il calcolo del valore di questa metrica è PragmaDB,

3.3.4.14 Test di sistema eseguiti

Lo strumento scelto per il calcolo del valore di questa metrica è PragmaDB,

3.3.4.15 Test di validazione eseguiti

Lo strumento scelto per il calcolo del valore di questa metrica è PragmaDB,

3.3.4.16 Test superati

Lo strumento scelto per il calcolo del valore di questa metrica è PragmaDB,

3.3.4.17 Completezza implementazione funzionale

Lo strumento scelto per il calcolo del valore di questa metrica è PragmaDB,

3.3.4.18 Densità di failure

Lo strumento scelto per il calcolo del valore di questa metrica è PragmaDB,

4 Processi organizzativi

4.1 Gestione

4.1.1 Scopo

Lo scopo del processo è produrre il “*Piano di Progetto v1.0.0*”, al fine di pianificare e gestire i ruoli che i membri dovranno assumere.

4.1.2 Aspettative

Le aspettative del processo sono:

- produrre il “*Piano di Progetto v1.0.0*”;
- definire i ruoli dei membri del gruppo;
- definire il piano per l'esecuzione dei compiti programmati.

4.1.3 Descrizione

4.1.4 Ruoli di progetto

In ogni momento temporale ogni membro deve ricoprire almeno un ruolo e, durante tutta la durata del *progetto*_g, ricoprire tutti i ruoli almeno una volta. Per ogni membro, le ore di lavoro devono essere il più possibile equamente distribuite. L'assegnazione e la rotazione dei ruoli sono pianificate nel “*Piano di Progetto v1.0.0*”.

4.1.4.1 Responsabile

Il *Responsabile* è il rappresentante e il punto di riferimento del gruppo, nonché colui che si assume le responsabilità delle scelte del gruppo. Le responsabilità assunte sono:

- pianificazione e coordinamento delle attività;
- analisi e gestione dei rischi;
- gestione delle risorse;
- approvazione dei documenti;
- approvazione dell'offerta economica;
- assicurarsi del rispetto delle “*Norme di Progetto v1.0.0*” e che vengano rispettate le pianificazioni nel “*Piano di Progetto v1.0.0*”.

4.1.4.2 Amministratore

L'*Amministratore* è responsabile dell'efficienza dell'ambiente di lavoro, in particolare si occupa di:

- studiare e fornire strumenti che migliorano l'ambiente di lavoro, automatizzando il lavoro ove possibile;
- gestire archiviazione, versionamento e configurazione dei documenti e del *software_g*;
- garantire la qualità del *prodotto_g*, fornendo procedure e strumenti di monitoraggio e segnalazione;
- eliminare le difficoltà sulla gestione di processi e risorse.

4.1.4.3 Analista

L'*Analista* deve identificare e comprendere il dominio del problema. In particolare si occupa di:

- mappare le richieste del cliente in specifiche per il *prodotto_g*;
- catalogare e spiegare specifiche comprensibili nell'“*Analisi dei Requisiti v1.0.0*” e nello “*Studio di Fattibilità v1.0.0*”.

4.1.4.4 Progettista

Il *Progettista* ha forti competenze sullo *stack_g* tecnologico usato. In particolare deve:

- indicare le tecnologie più adatte allo sviluppo del *progetto_g*;
- descrivere il funzionamento del *sistema_g* progettandone l'architettura;
- produrre una soluzione fattibile in termini di risorse.

4.1.4.5 Programmatore

Il *Programmatore* si occupa della codifica, in particolare:

- implementa le soluzioni indicate dal *Progettista*;
- scrive codice documentato, versionato e mantenibile nel rispetto delle “*Norme di Progetto v1.0.0*”;
- realizza e fornisce gli strumenti per verificare e validare il *prodotto_g*.

4.1.4.6 Verificatore

Il *Verificatore*, disponendo di una profonda conoscenza delle “*Norme di Progetto v1.0.0*”, si occupa delle attività di *verifica_g*.

In particolare deve:

- controllare il rispetto delle “*Norme di Progetto v1.0.0*” durante ogni attività del *progetto_g*.

4.1.5 Comunicazioni

4.1.5.1 Interne

È stato creato un gruppo Telegram, accessibile solo ai membri del team, per effettuare le comunicazioni interne. In caso siano necessaria maggiore interazione, si farà utilizzo di *Google Hangouts_g*.

4.1.5.2 Esterne

È stata creata un'apposita cartella di posta elettronica per mantenere i contatti con il *proponente_g*, il committente ed altre eventuali figure esterne. La gestione della casella di posta elettronica è compito del *Responsabile*.

L'indirizzo e-mail è il seguente: swe.co.code@gmail.com.

4.1.6 Incontri

4.1.6.1 Interni

Ogni membro del team può proporre un incontro interno tramite il *bot_g* Telegram "*VotePoll_g*", specificando i motivi e l'oggetto dell'incontro. Sarà poi compito del *Responsabile* decidere se effettuare l'incontro o meno.

La verbalizzazione degli incontri interni è compito di uno tra gli *Amministratori*.

4.1.6.2 Esterni

Ogni membro del team può proporre un incontro esterno tramite il *Bot_g* Telegram "*VotePoll_g*", specificando i motivi e l'oggetto dell'incontro. Se il *Responsabile* decide che l'incontro può essere organizzato dovrà accordarsi con la figura esterna, e comunicare gli estremi della riunione ai membri del team.

La verbalizzazione degli incontri esterni è compito del *Responsabile*.

4.1.7 Strumenti di coordinamento

4.1.7.1 Ticketing_g

Il *Responsabile* ha il compito di assegnare i *task_g* ai membri del team utilizzando l'applicativo web *Asana_g*.

Definendo delle *milestone_g*, è possibile tenere traccia dello stato di avanzamento del lavoro di ogni *task_g*.

4.1.8 Strumenti di versionamento

4.1.8.1 Repository_g

Per il versionamento e l'archiviazione dei file, l'*Amministratore* ha creato un *repository_g* *GitHub_g*, il quale è disponibile al seguente indirizzo <https://github.com/CoCodeSWE>. Tutti i membri del gruppo dovranno creare un proprio account *GitHub_g*, per poi ricevere i permessi in scrittura sul *repository_g* da parte dell'*Amministratore*. La gestione del *repository_g* è responsabilità degli *Amministratori*.

4.1.8.2 Struttura del *repository*_g Docs

Al fine di mantenere ordine e coerenza tra i file, il *repository*_g è così strutturato:

- Docs
 - RR
 - * Esterni: contiene i documenti esterni;
 - * Interni: contiene i documenti interni.
 - script: contiene gli script utilizzati;
 - *template*_g: contiene i *template*_g utilizzati.

In futuro verranno aggiunte nuove *repository*_g, le quali strutture saranno rappresentate come precedentemente fatto con Docs.

4.1.8.3 Commit

Ogni commit effettuata deve essere accompagnata da un messaggio descrittivo delle modifiche effettuate. L'autore della commit dovrà assicurarsi della correttezza dei file. È sconsigliato effettuare il commit di intere cartelle, al fine di evitare inclusioni di file inutili (ad es: file di compilazione). Dovrà inoltre essere segnalata l'eventuale aggiunta di nuovi file.

4.1.9 Rischi

Il *Responsabile* ha il dovere di individuare e monitorare i rischi indicati nel “*Piano di Progetto v1.0.0*”. In caso ne vengano identificati di nuovi, il *Responsabile* deve agire nel modo seguente:

- comunicare i nuovi rischi al team;
- pianificare una strategia per la gestione dei nuovi rischi;
- aggiornare le procedure di gestione dei rischi nel “*Piano di Progetto v1.0.0*”.

4.1.10 Strumenti

4.1.10.1 Telegram

Telegram è un *software*_g libero che fornisce un servizio di messaggistica istantanea erogato senza fini di lucro dalla società Telegram LLC. È stato ritenuto più adatto di Whatsapp.

4.1.10.2 Google Hangouts_g

Hangouts è un *software*_g di messaggistica istantanea e di *VoIP*_g sviluppato da Google. È disponibile per le piattaforme mobili *Android*_g e *iOS*_g e come estensione per il *browser*_g web Google Chrome. Inoltre, permette la condivisione degli schermi tra i membri della chiamata. È stato ritenuto più adatto di Skype.

4.1.10.3 Git_g

Git_g è un *software*_g open-source di controllo versione distribuito utilizzabile dal terminale. Come versione si utilizza la 2.7.4 o superiori.

4.1.10.4 *GitHub_g*

GitHub_g è un servizio di hosting per progetti *software_g*, con il quale è possibile interagire tramite *Git_g*. *GitHub_g* offre diversi piani per *repository_g* privati sia a pagamento, sia gratuiti, molto utilizzati per lo sviluppo di progetti open-source.

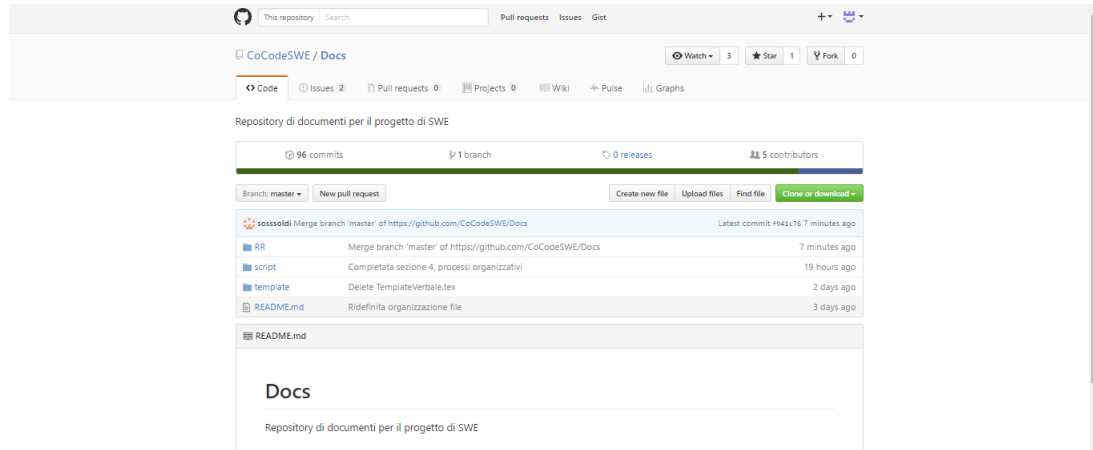


Figura 4: Github

4.1.10.5 *GitHub_g* desktop

GitHub_g Desktop è l'applicativo desktop per contribuire e collaborare ai progetti del corrispondente servizio web *GitHub_g*. Esso è disponibile per Windows e MacOS. Per Windows si utilizza la versione 3.3.3 o superiori.

4.1.10.6 *Asana_g*

Asana_g è un applicativo web e mobile che consente al team di assegnare, tracciare e gestire dei *task_g*.

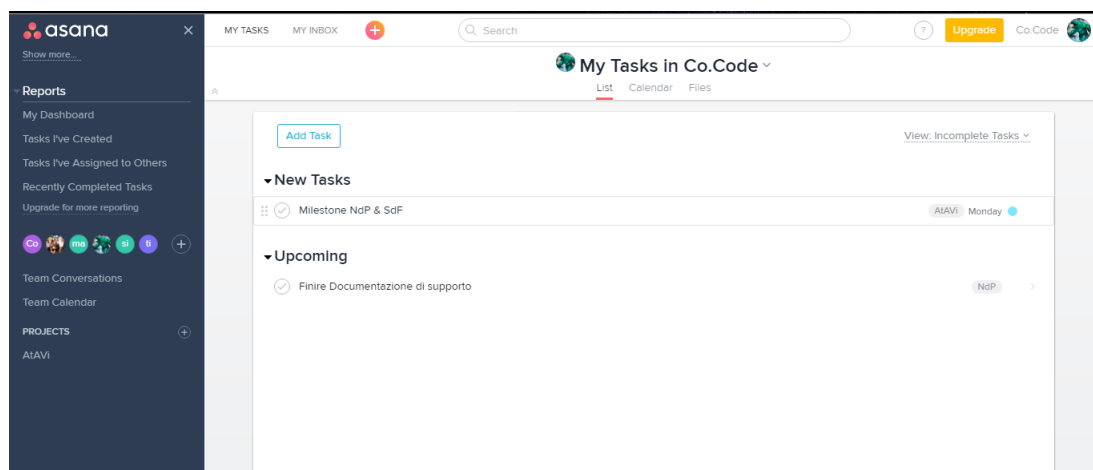


Figura 5: Asana

4.1.10.7 GanttProject

GanttProject è un *software*_g gratuito per la creazione di grafici rappresentanti l'organizzazione e gestione di compiti e *milestone*_g all'interno di un *progetto*_g. Verrà utilizzato nella versione 2.7 o superiore.

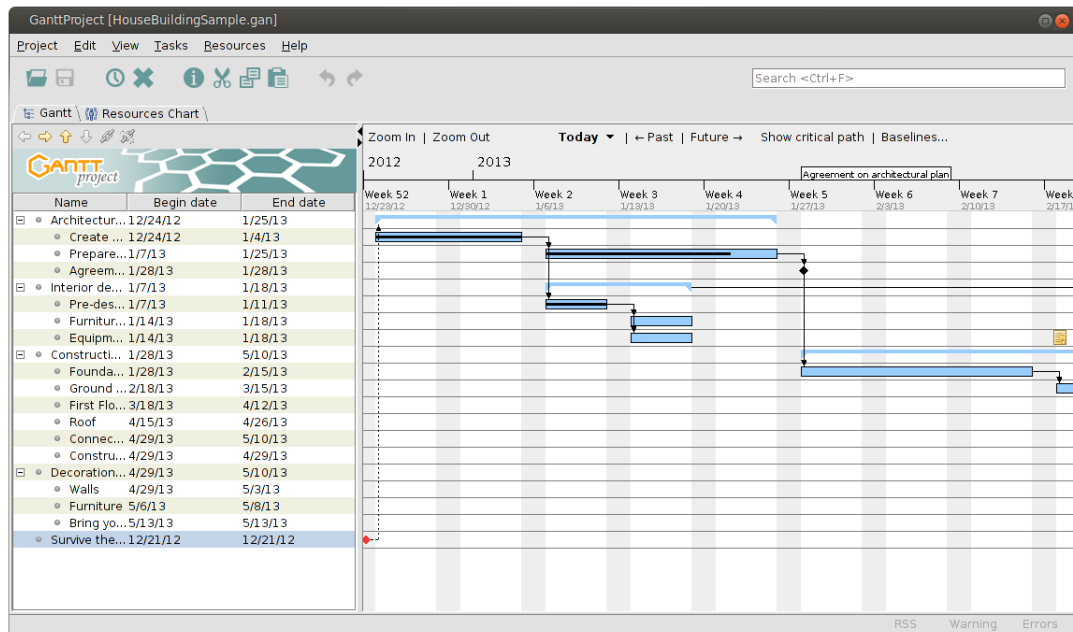


Figura 6: GanttProject