



AtAVi

Norme di *progetto*_g v1.0

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
Data di redazione	2016-12-10
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
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	5
1.1	Scopo del documento	5
1.2	Scopo del <i>prodotto_g</i>	5
1.3	Glossario	5
1.4	Riferimenti	5
1.4.1	Riferimenti Normativi	5
1.4.2	Riferimenti Informativi	5
2	Processi primari	6
2.1	Sviluppo	6
2.1.1	Scopo	6
2.1.2	Aspettative	6
2.1.3	Descrizione	6
2.1.4	Analisi dei requisiti	6
2.1.4.1	Scopo dell'attività	6
2.1.4.2	Aspettative dell'attività	6
2.1.4.3	Descrizione dell'attività	6
2.1.4.4	Studio di fattibilità	7
2.1.4.5	Casi d'uso	7
2.1.4.6	Codice identificativo dei casi d'uso	7
2.1.4.7	Requisiti	7
2.1.4.8	Codice identificativo dei requisiti	7
2.1.4.9	<i>UML_g</i>	8
2.1.5	Progettazione	8
2.1.5.1	Scopo dell'attività	8
2.1.5.2	Aspettative dell'attività	8
2.1.5.3	Descrizione dell'attività	8
2.1.5.4	Specifica tecnica	8
2.1.5.5	Definizione di <i>prodotto_g</i>	8
2.1.6	Codifica	9
2.1.6.1	Scopo dell'attività	9
2.1.6.2	Aspettative dell'attività	9
2.1.6.3	Descrizione dell'attività	9
2.1.6.4	Stile	9
2.1.6.5	Versionamento	9
2.1.6.6	<i>Ricorsione_g</i>	9
2.1.7	Strumenti	9
2.1.7.1	Astah	9
2.1.7.2	PragmaDB	10
3	Processi di supporto	10
3.1	Documentazione	10
3.1.1	Scopo	10
3.1.2	Aspettative	10
3.1.3	Descrizione	10
3.1.4	Procedure	10
3.1.4.1	Approvazione dei documenti	10
3.1.5	Template	10
3.1.6	Struttura dei documenti	11
3.1.6.1	Frontespizio	11
3.1.6.2	Diario delle modifiche	11
3.1.6.3	Indice	11
3.1.6.4	Intestazione e piè di pagina	11

3.1.7	Versionamento	12
3.1.8	Norme tipografiche	12
3.1.8.1	Stile del testo	12
3.1.8.2	Elenchi puntati	13
3.1.8.3	Formati comuni	13
3.1.8.4	Sigle	13
3.1.9	Elementi grafici	14
3.1.9.1	Tabelle	14
3.1.9.2	Immagini	14
3.1.10	Classificazione dei documenti	14
3.1.10.1	Documenti informali	14
3.1.10.2	Documenti formali	14
3.1.10.3	Glossario	14
3.1.10.4	Verbalì	15
3.1.11	Strumenti	15
3.1.11.1	L ^A T _E X	15
3.1.11.2	TexMaker	16
3.2	Verifica	16
3.2.1	Scopo del processo	16
3.2.2	Aspettative del processo	16
3.2.3	Attività	16
3.2.3.1	Analisi statica	16
3.2.3.2	Analisi dinamica	16
3.2.4	Issue tracking	16
3.2.4.1	Gestione delle issue	17
3.2.5	Strumenti	17
3.2.5.1	Strumenti per l'issue tracking	17
3.2.5.2	Verifica ortografica	17
4	Processi organizzativi	17
4.1	Gestione	17
4.1.1	Scopo	17
4.1.2	Aspettative	17
4.1.3	Descrizione	17
4.1.4	Ruoli di progetto	17
4.1.4.1	Responsabile	18
4.1.4.2	Amministratore	18
4.1.4.3	Analista	18
4.1.4.4	Progettista	18
4.1.4.5	Programmatore	18
4.1.4.6	Verificatore	19
4.1.5	Comunicazioni	19
4.1.5.1	Interne	19
4.1.5.2	Esterne	19
4.1.6	Incontri	19
4.1.6.1	Interni	19
4.1.6.2	Esterni	19
4.1.7	Strumenti di coordinamento	19
4.1.7.1	Ticketing	19
4.1.8	Strumenti di versionamento	20
4.1.8.1	<i>Repository_g</i>	20
4.1.8.2	Struttura del <i>repository_g</i>	20
4.1.8.3	Commit	20
4.1.9	Rischi	20
4.1.10	Strumenti	20
4.1.10.1	Telegram	20

4.1.10.2	Google Hangout	21
4.1.10.3	Git	21
4.1.10.4	GitHub	21
4.1.10.5	GitHub desktop	21
4.1.10.6	Asana	21
4.1.10.7	strumento per i diagrammi di gantt	21

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 dell'arrivo dell'ospite sul sistema di comunicazione aziendale (Slack) e fornire un'attività di accoglienza.

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 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

2 Processi primari

2.1 Sviluppo

2.1.1 Scopo

Include le attività e i compiti svolti per produrre 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.

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 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). Il tracciamento dei requisiti avviene tramite ...

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 a cui si può incorrere nel *capitolato_g* in analisi.

2.1.4.5 Casi d'uso

Ogni caso d'uso è così composto:

- codice identificativo;
- titolo;
- diagramma *UML_g*;
- attori primari;
- attori secondari;
- scopo;
- descrizione;
- preconditione;
- scenario principale;
- scenari alternativi.

2.1.4.6 Codice identificativo dei casi d'uso

da decidere

2.1.4.7 Requisiti

Ogni requisito è così composto:

- codice identificativo;
- tipologia;
- descrizione;
- fonti.

2.1.4.8 Codice identificativo dei requisiti

da decidere

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 sopra citati, 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. Trender????
- **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

Lo stile di codifica verrà trattato e descritto 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 Astah

Astah è uno strumento di modellazione UML. Verrà quindi usato per la produzione di diagrammi UML. Viene utilizzata la versione 7.0 o superiori.

2.1.7.2 PragmaDB

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 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 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 4.1.11.

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. il *Responsabile* di *progetto_g* consegna il documento ai *Verificatori*, i quali dovranno occuparsi di controllare la correttezza dello stesso;
3. se i *Verificatori* riscontrano degli errori si ritorna al punto 1, altrimenti 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.

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 da una tabella ordinata in modo decrescente secondo la data di modifica e il numero di versione.

Gli *attributi_g* della tabella rappresentano:

- numero di versione;
- breve riepilogo delle modifiche apportate;
- autore delle modifiche;
- ruolo ricoperto dall'attore 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;
 - è sempre minore del numero totali di revisioni.
- **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 `LATEXcreate`). 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 due cifre;
- GG: rappresenta il giorno usando due 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:** ogni nome di un ruolo di progetto 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 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, 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. Solo i documenti formali possono essere distribuiti all’esterno del gruppo. Per arrivare a tale stato il documento deve aver già passato la verifica e la validazione.

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 debba generare 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_DataVerbale

dove:

- 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:

DX.Y

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 al posto della D viene messa una S.

3.1.11 Strumenti

3.1.11.1 L^AT_EX

La stesura dei documenti deve essere effettuata utilizzando il linguaggio di markup L^AT_EX. Le motivazioni di questa scelta sono dovute alle possibilità che L^AT_EX 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 separato e condiviso da tutti i documenti;
- creazione e gestione automatica dell'indice del documento.

Inoltre è stato reso disponibile uno script PHP che per ogni documento L^AT_EX 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

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

3.2 Verifica

3.2.1 Scopo del processo

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

3.2.2 Aspettative del processo

Una corretta implementazione di tale processo permette di individuare:

- una procedura di verifica;
- i criteri per la verifica del prodotto.

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 non è ancora disponibile e durante tutto l'arco del suo sviluppo. Non richiede l'esecuzione del prodotto software 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, 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 e 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 applicabile solamente al software. Tale tecnica può essere utilizzata per analizzare l'intero software 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 del software.

3.2.4 Issue tracking

L'issue tracking è un'attività di supporto per la figura del *Verificatore*, ai quali permette di tenere traccia, e contemporaneamente segnalare al *Responsabile*, la presenza di potenziali errori in un documento o in nel codice sorgente.

3.2.4.1 Gestione delle issue

Qualora un *Verificatore* dovesse riscontrare delle anomalie, la procedura per la segnalazione e gestione del ticketing di una issue è la seguente:

1. il *Verificatore* dovrà aprire una nuova issue assegnandole una label che si riferisca al problema trovato;
2. il *Responsabile* di progetto dovrà valutare la issue; se la ritiene essere appropriata assegnerà ai redattori del documento (o ai *Programmatore*) il compito di risolvere la issue;
3. una volta risolta, e verificata, la issue dovrà essere marcata come conclusa;

3.2.5 Strumenti

3.2.5.1 Strumenti per l'issue tracking

Lo strumento utilizzato per l'issue tracking è il servizio Issues messo a disposizione da GitHub.

3.2.5.2 Verifica ortografica

Viene utilizzata la verifica in tempo reale dell'ortografia, integrata in TexMaker. Essa marca, sottolineando in rosso, le parole errate secondo la lingua italiana.

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 Hangout.

4.1.5.2 Esterne

È stata creata un'apposita cartella di posta elettronica per mantenere i contatti con il proponente, 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 Telegram “VotePoll”, specificando i motivi e l'oggetto dell'incontro. Sarà poi compito del *Responsabile* decidere se effettuare l'incontro o meno.

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

4.1.6.2 Esterni

Ogni membro del team può proporre un incontro esterno tramite il Bot Telegram “VotePoll”, specificando i motivi e l'oggetto dell'incontro. Sarà poi compito del *Responsabile* decidere se organizzare l'incontro o meno. Una volta che il *Responsabile* si è accordato con la figura esterna, egli dovrà comunicare gli estremi dell'incontro ai membri del team.

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

4.1.7 Strumenti di coordinamento

4.1.7.1 Ticketing

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

Definendo delle milestone, è possibile tenere traccia dello stato di avanzamento del lavoro di ogni task.

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 GitHub, il quale è disponibile al seguente indirizzo <https://github.com/CoCodeSWE/Docs>. Tutti i membri del gruppo dovranno creare un proprio account GitHub, per poi ricevere i permessi in scrittura sul repository da parte dell'*Amministratore*. La gestione del repository è responsabilità degli *Amministratori*.

4.1.8.2 Struttura del *repository_g*

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

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

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 committare 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 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 Hangout

Hangouts è un software di messaggistica istantanea e di VoIP sviluppato da Google. È disponibile per le piattaforme mobili Android e iOS e come estensione per il browser 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

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

4.1.10.4 GitHub

GitHub è un servizio di hosting per progetti software, con il quale è possibile interagire tramite Git. GitHub offre diversi piani per repository privati sia a pagamento, sia gratuiti, molto utilizzati per lo sviluppo di progetti open-source.

4.1.10.5 GitHub desktop

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

4.1.10.6 Asana

Asana è un applicativo web e mobile che consente al team di assegnare, tracciare e gestire dei task.

4.1.10.7 strumento per i diagrammi di gantt