



Agents of S.W.E.

A SOFTWARE COMPANY

Agents of S.W.E - Progetto "Plugin Grafana(?)"

Norme di Progetto

Versione	0.0.2
Approvazione	?
Redazione	Luca Violato Bogdan Stanciu Marco Favaro Marco Chilese
Verifica	? ?
Stato	Work in Progress
Uso	Interno
Destinato a	Agents of S.W.E Prof. Tullio Vardanega Prof. Riccardo Cardin

agentsofswe@gmail.com

Indice

1	Introduzione	1
1.1	Scopo del Documento	1
1.2	Ambiguità e Glossario	1
1.3	Riferimenti	1
2	Processi Primari	2
2.1	Fornitura	2
2.2	Sviluppo	2
2.2.1	Analisi dei requisiti	2
2.2.1.1	Classificazione dei requisiti	2
2.2.1.2	Classificazione dei casi d'uso	2
2.2.2	Progettazione	2
2.2.3	Codifica	2
2.2.3.1	Convenzioni per i nomi:	2
2.2.3.2	Convenzioni per la documentazione:	2
3	Processi di supporto	4
3.1	Versionamento	4
3.1.1	Controllo di versione	4
3.1.1.1	Struttura del repository	4
3.1.1.2	Processo di implementazione	4
3.1.1.3	Ciclo di vita dei branch	5
3.1.1.4	Rilascio di versione	5
3.1.2	Configurazione versionamento	5
3.1.2.1	Remoto	5
3.1.2.2	Locale	5
3.2	Gestione di progetto	5
3.2.1	Configurazione strumenti di organizzazione	5
3.2.1.1	Inizializzazione	5
3.2.1.2	Aggiunta tasks	5
3.2.1.3	Aggiunta milestones	6
3.2.2	Ciclo di vita delle tasks	6
3.2.2.1	Apertura	6
3.2.2.2	Completamento	6
3.2.2.3	Richiesta di revisione	6
3.2.2.4	Chiusura	6

3.2.3	Strumenti	6
3.3	Verifica	6
3.3.1	Descrizione	6
3.3.2	Analisi Statica	6
3.3.3	Verifica Diagrammi UML	6
3.3.4	Strumenti	6
3.4	Validazione	6
3.4.1	Descrizione	6
3.4.2	Procedure	6
3.5	Risoluzione problematiche	6
3.5.1	Descrizione	6
3.5.2	Ciclo di vita	6
3.5.2.1	Individuazione	6
3.5.2.2	Analisi problematiche	6
3.5.2.3	Scelta soluzione migliore	6
3.5.2.4	Reinserimento nella gestione di processo	7
4	Processi Organizzativi	7
4.1	Gestione	7
5	Changelog	8

1 Introduzione

1.1 Scopo del Documento

1.2 Ambiguità e Glossario

1.3 Riferimenti

2 Processi Primari

2.1 Fornitura

2.2 Sviluppo

2.2.1 Analisi dei requisiti

2.2.1.1 Classificazione dei requisiti

2.2.1.2 Classificazione dei casi d'uso

2.2.2 Progettazione

2.2.3 Codifica

Di seguito vengono definite delle norme che devono essere adottate dai Programatori per garantire una buona leggibilità e manutenibilità del codice. Le prime norme che seguiranno sono le più generali, da adottarsi per ogni linguaggio di programmazione adottato all'interno del progetto, in seguito quelle più specifiche per i linguaggi JavaScript_G, HTML_G e CSS_G.

Ogni norma è caratterizzata da un paragrafo di appartenenza, da un titolo, una breve descrizione, e se il caso lo richiede, un esempio.

Il rispetto delle seguenti norme è fondamentale per garantire uno stile di codifica uniforme all'interno del progetto, oltre che per massimizzare la leggibilità e agevolare la manutenzione, la verifica_G e la validazione_G.

2.2.3.1 Convenzioni per i nomi:

- I Programatori devono adottare come notazione per la definizione di cartelle, file, metodi, funzioni e variabili il CamelCase_G.

Di seguito un esempio di corretta nomenclatura:

INSERIRE ESEMPIO

- Tutti i nomi devono essere **unici** ed **autoesplicativi**, ciò per evitare ambiguità e limitare la complessità.

2.2.3.2 Convenzioni per la documentazione:

- Tutti i nomi ed i commenti al codice vanno scritti in **inglese**;

- Nel codice è possibile utilizzare un commento con denominazione **TODO** in cui si vanno ad indicare compiti da svolgere;
- L'intestazione di ogni file deve essere la seguente:

INSERIRE INTESTAZIONE FILE

- La versione del file nell'intestazione, deve rispettare la seguente formulazione: $Y.K$, dove Y rappresenta la versione principale, K la versione parziale della relativa versione principale.

I numeri di versione del tipo $Y.0$, dalla 1.0, vengono considerate versioni stabili, e quindi versioni da testare per saggiarne la qualità.

3 Processi di supporto

3.1 Versionamento

La necessità di più componenti del **gruppo** di cooperare su uno stesso documento, porta alla soluzione di utilizzare un sistema di versionamento distribuito.

3.1.1 Controllo di versione

Il sistema di versionamento, utilizzato in questa fase di **RR**, è **git**, con il supporto hosting di **GitHub**.

3.1.1.1 Struttura del repository La struttura del repository segue il workflow **gitflow** di **Driessen at nvie**, idealizzato attorno il concetto di **release** del prodotto. Questo produce un framework robusto attorno al quale si possono gestire progetti di grandi dimensioni. I due branch principali sono il **master** e in parallelo ad esso il **develop**. Il **master** viene considerato il branch *main*, dove il codice sorgente della *testa* riflette sempre lo stato di "*production-ready*", mentre il ramo di **develop** è considerato il branch principale dove vengono effettuate le ultime modifiche per il prossimo rilascio del prodotto.

3.1.1.2 Processo di implementazione L'implementazione dei documenti avviene tramite gli strumenti utilizzati nel paragrafo –INSERIRE PARAGRAFO PARTE DI FAVARO–

Quest'ultimi, in fase di compilazione producono dei file di poca rilevanza con estensioni come: *.log*, *.out*, *.idx*, *.aux*, *.gz*, *.aux*, *.toc*, i quali verranno ignorati come da configurazione, ma soprattutto file di più rilevanza come *.pdf*, *.tex*, i quali verranno versionati dal sistema di git preinstallato. Una volta creati/modificati i documenti, si procede con il `git` commit di essi. Il commit riporta un *cambiamento* al file, con un messaggio allegato ad esso che ne descrive le modifiche apportate o un comando apposito per chiudere alcune task con tale commit. Dopo di che, il commit viene `git` pushato nel branch appropriato, a seconda dei criteri descritti nel prossimo paragrafo.

3.1.1.3 Ciclo di vita dei branch

1. **Master**: branch *main* del repository, esso rappresenta lo stato di "*production-ready*" del prodotto. Questo branch ha una durata di vita quanto il repository stesso o infinita;
2. **Develop**: branch di sviluppo parallelo al **master** sul quale vengono aggiunte le feature provenienti appunto dai branch **feature**, e dal quale inizia il branch di **release**. Ha una durata di vita quanto il branch master;
3. **Release**: branch di preparazione per un nuovo rilascio o aggiornamento del prodotto. Utilizzato per risolvere piccoli errori e configurare le impostazioni di rilascio. Una volta rilasciato il prodotto, esso si riversa sul branch **master** e **develop**. Ha una durata breve in quanto il rilascio deve essere effettuato il prima possibile;
4. **Feature**: branch usato per sviluppare nuove feature per il prossimo rilascio a breve o lungo tempo. Il suo tempo di vita dura quanto lo sviluppo della nuova feature fintanto che non avviene il G merge sul branch di **develop**;
5. **Hotfix**: branch molto simili a quelli di release, con l'obiettivo di risolvere immediatamente un bug del prodotto in produzione o release. Una volta risolto il bug, esso si riversa sui branch **master** e **develop**, aggiornandoli nel minor tempo possibile. Ha un tempo di vita breve, in quanto viene creato per la necessità di risolvere un problema sul prodotto rilasciato.

3.1.1.4 Rilascio di versione

3.1.2 Configurazione versionamento

3.1.2.1 Remoto

3.1.2.2 Locale

3.2 Gestione di progetto

3.2.1 Configurazione strumenti di organizzazione

3.2.1.1 Inizializzazione

3.2.1.2 Aggiunta tasks

3.2.1.3 Aggiunta milestones**3.2.2 Ciclo di vita delle tasks****3.2.2.1 Apertura****3.2.2.2 Completamento****3.2.2.3 Richiesta di revisione****3.2.2.4 Chiusura****3.2.3 Strumenti****3.3 Verifica****3.3.1 Descrizione****3.3.2 Analisti Statica****3.3.3 Verifica Diagrammi UML****3.3.4 Strumenti****3.4 Validazione****3.4.1 Descrizione****3.4.2 Procedure****3.5 Risoluzione problematiche****3.5.1 Descrizione****3.5.2 Ciclo di vita****3.5.2.1 Individuazione****3.5.2.2 Analisi problematiche****3.5.2.3 Scelta soluzione migliore**

3.5.2.4 Reinserimento nella gestione di processo

4 Processi Organizzativi

4.1 Gestione

5 Changelog

Versione	Data	Autore	Ruolo	Descrizione
0.0.1	2018-11-23	Luca Violato	?	Strutturazione del Documento
0.0.2	2018-11-23	Marco Chilese	Verificatore	Prima stesura sezione Processi Primari
0.0.3	2018-12-03	Bogdan Stanciu	Responsabile	Stesura processi di supporto

Tabella 1: Changelog del documento

Versione	Data	Autore	Ruolo	Descrizione
0.0.1	2018-11-23	Luca Violato	?	Strutturazione del Documento
0.0.2	2018-11-23	Marco Chilese	Verificatore	Prima stesura sezione Processi Primari
0.0.3	2018-12-03	Bogdan Stanciu	Responsabile	Stesura processi di supporto