



Norme di Progetto

SWEefty - 4 Dicembre 2017

Informazioni sul documento

Versione	0.0
Redazione	Alberto Galinaro Davide Zago Elia Montecchio Francesco Parolini Giuseppe Merlino Lisa Parma Paolo Eccher
Verifica	XXX
Approvazione	YYY
Uso	Interno/esterno
Distribuzione	ZZZ

Descrizione

Questo documento descrive le regole, gli strumenti e le convenzioni adottate dal gruppo SWEefty durante la realizzazione del progetto etctectetc.

Diario delle modifiche

Modifica	Autore	Ruolo	Data	Versione
<i>Prima stesura dello sche- letro del documento</i>	Francesco Parolini	Project manager	2017-12-04	0.01
<i>Scrittura della sezione "Fornitura" dei processi primari</i>	Alberto Gallinaro	Chief Artist	2017-12-07	0.02
<i>Scrittura della sezione "Codifica" dei processi primari</i>	Alberto Gallinaro	Chief Artist	2017-12-07	0.03

Indice

1	Introduzione	3
1.1	Scopo del documento	3
1.2	Scopo del prodotto	3
1.3	Ambiguità	3
1.4	Riferimenti	3
1.4.1	Normativi	3
1.4.2	Informatici	3
2	Processi primari	3
2.1	Fornitura	3
2.1.1	Scopo	3
2.1.2	Aspettative	3
2.1.3	Descrizione	4
2.1.4	Attività	4
2.1.4.1	Studio di fattibilità	4
2.1.4.2	Piano di Progetto	4
2.1.4.3	Piano di Qualifica	4
2.2	Sviluppo	4
2.2.1	Scopo	4
2.2.2	Aspettative	5
2.2.3	Descrizione	5
2.2.4	Attività	5
2.2.4.1	Analisi dei requisiti	5
2.2.4.1.1	Scopo	5
2.2.4.1.2	Aspettative	5
2.2.4.1.3	Descrizione	5
2.2.4.1.4	Casi d'uso	5
2.2.4.1.5	Codice identificativo	5
2.2.4.1.6	Requisiti	5
2.2.4.1.7	Codice identificativo	6
2.2.4.1.8	UML	6
2.2.4.2	Progettazione	6
2.2.4.2.1	Scopo	6
2.2.4.2.2	Aspettative	6
2.2.4.2.3	Descrizione	6
2.2.4.2.4	Specifica Tecnica	6
2.2.4.2.5	Definizione di Prodotto	6
2.2.4.3	Codifica	6
2.2.4.3.1	Scopo	6
2.2.4.3.2	Aspettative	6
2.2.4.3.3	Descrizione	6
2.2.4.3.4	Stile di codifica	6
2.2.4.3.5	Intestazione	8
2.2.4.3.6	Versionamento	8
2.2.4.3.7	Ricorsione	8
2.2.5	Strumenti	8
2.2.5.1	Trender	8
2.2.5.2	Astah	8
2.2.5.3	IntelliJ IDEA	8
3	Processi Organizzativi	8
3.1	Gestione	8
3.1.1	Scopo	8
3.1.2	Aspettative	8

3.1.3	Descrizione	8
3.1.4	Ruoli di progetto	8
3.1.4.1	Amministratore di Progetto	8
3.1.4.2	Responsabile di Progetto	8
3.1.4.3	Analista	9
3.1.4.4	Progettista	9
3.1.4.5	Verificatore	9
3.1.4.6	Programmatore	9
3.1.5	Procedure	9
3.1.5.1	Gestione delle comunicazioni	9
3.1.5.1.1	Comunicazioni interne	9
3.1.5.1.2	Comunicazioni esterne	9
3.1.5.2	Gestione degli incontri	9
3.1.5.2.1	Incontri interni	9
3.1.5.2.2	Incontri esterni	9
3.1.5.3	Gestione degli strumenti di coordinamento	9
3.1.5.3.1	Ticketing	9
3.1.5.4	Gestione degli strumenti di versionamento	9
3.1.5.4.1	Repository	9
3.1.5.4.2	Struttura del repository	9
3.1.5.4.3	Tipi di file e .gitignore	9
3.1.5.4.4	Norme sui commit	9
3.1.5.5	Gestione dei rischi	9
3.1.6	Strumenti	9
3.1.6.1	Sistema Operativo	9
3.1.6.2	Slack	9
3.1.6.3	Telegram	10
3.1.6.4	Wrike	10
3.1.6.5	Git	10
3.1.6.6	Github	10
4	Processi di Supporto	10
4.1	Documentazione	10
4.1.1	Scopo	10
4.1.2	Aspettative	10
4.1.3	Descrizione	10
4.1.4	Procedure	10
4.1.4.1	Approvazione dei documenti	10
4.1.5	Template	10
4.1.6	Struttura dei documenti	10
4.1.6.1	Prima pagina	10
4.1.6.2	Registro delle modifiche	10
4.1.6.3	Indice	10
4.1.6.4	Contenuto principale	10
4.1.6.5	Note a piè di pagina	10
4.1.7	Versionamento	10
4.1.8	Norme tipografiche	10
4.1.8.1	Stile del testo	10
4.1.8.2	Elenchi puntati	10
4.1.8.3	Formati comuni	10
4.1.8.4	Sigle	11
4.1.9	Elementi grafici	11
4.1.9.1	Tabelle	11
4.1.9.2	Immagini	11
4.1.10	Classificazione dei documenti	11
4.1.10.1	Documenti informali	11

4.1.10.2	Documenti formali	11
4.1.10.3	Verbali	11
4.1.11	Strumenti	11
4.1.11.1	L ^A T _E X	11
4.1.11.2	TexStudio	11
4.1.11.3	Lucidchart	11
4.2	Verifica	11
4.2.1	Scopo	11
4.2.2	Aspettative	11
4.2.3	Descrizione	11
4.2.4	Attività	11
4.2.4.1	Analisi	11
4.2.4.1.1	Analisi Statica	11
4.2.4.1.2	Analisi dinamica	11
4.2.4.2	Test	11
4.2.4.2.1	Test di unità	11
4.2.4.2.2	Test di integrazione	11
4.2.4.2.3	Test di sistema	12
4.2.4.2.4	Test di regressione	12
4.2.4.2.5	Test di accettazione	12
4.2.5	Strumenti	12
4.2.5.1	Verifica ortografica	12
4.2.5.2	Validazione W3C	12
4.2.5.3	Analisi statica	12
4.2.5.4	Analisi dinamica	12
4.2.5.5	Metriche	12

1 Introduzione

1.1 Scopo del documento

Il seguente documento ha l'obiettivo di esplicitare le norme, le convenzioni e la strumentazione che sarà adottata dal gruppo SWEefty durante l'intero svolgimento del progetto. Con questa prospettiva deve essere visionato da tutti i membri del gruppo, i quali sono tenuti ad osservare quanto scritto per mantenere consistenza ed omogeneità in ogni aspetto del ciclo di vita del software che sarà prodotto durante il progetto.

1.2 Scopo del prodotto

Lo scopo del *prodotto_G* è sviluppare dei plugin Kibana in attraverso la libreria Javascript D3.

1.3 Ambiguità

Per scongiurare ogni malinteso ed ogni ambiguità nella terminologia impiegata viene fornito il *Glossario*, il quale contiene le definizioni dei termini in corsivo con una G a pedice.

1.4 Riferimenti

1.4.1 Normativi

- Verbale di incontro interno:
- Verbale di incontro esterno:

1.4.2 Informatici

eventuali guide latex, git, però devono essere serie....

2 Processi primari

2.1 Fornitura

2.1.1 Scopo

Lo scopo di questo *processo_G* è di trattare i termini e le norme, dalle piu' triviali alle piu' importanti, che tutti i componenti del gruppo SWEefty sono tenuti a rispettare per diventare fornitori dell'azienda IKS s.r.l e dei committenti Prof. Tullio Vardanega e Prod. Riccardo Cardin

2.1.2 Aspettative

Nel corso dell'intero progetto il gruppo intende instaurare con IKS in particolare nelle figure dei referente Stefano Bertolin e Stefano Lazzaro un rapporto di costante collaborazione al fine di:

- Stimare i costi del prodotto finale
- Concordare la qualifica del prodotto
- Determinare vincoli su processi e requisiti
- Determinare aspetti fondamentali al fine di soddisfare la necessità del proponente

2.1.3 Descrizione

Intendiamo mantenere un costante rapporto con il proponente in modo tale da ricevere un *feedback_G* costante sul lavoro svolto.

2.1.4 Attività

2.1.4.1 Studio di fattibilità

Sono state organizzate diverse riunioni al fine di permettere ai membri del team di scambiarsi opinioni sui capitolati proposti. Abbiamo valutato il capitolato secondo diversi criteri:

- **Dominio tecnologico:** E' stata presa in considerazione la conoscenza attuale delle tecnologie richieste per portare a termine i vari capitolati, in base anche a esperienze passate avute con problemi simili;
- **Individuazione rischi:** sono state analizzate le varie difficoltà che potrebbero essere incontrare durante la realizzazione dei vari capitolati considerando in modo particolare la mancanza di conoscenze adeguate relative alle tecnologie necessarie per realizzare i capitolati.

2.1.4.2 Piano di Progetto

Il *Project Manager* affiancato dagli *Amministratori* dovrà stilare un piano da seguire per la realizzazione del progetto. Il documento dovrà rispettare e seguire i seguenti punti:

- **Analisi dei rischi:** si analizzano nel dettaglio i rischi che si potrebbero presentare durante lo svolgimento del progetto, capendo con quale probabilità potrebbero accadere e qual è il livello di gravità di ogni rischio;
- **Pianificazione:** si pianificano le attività da svolgere nel corso del progetto fornendo delle scadenze temporali precise;
- **Preventivo:** si stima, secondo la pianificazione, la quantità di lavoro necessaria per portare a termine ogni fase, in modo tale da riuscire a proporre un preventivo finale per il costo totale del progetto.

2.1.4.3 Piano di Qualifica

Ai *Verificatori* è assegnato il compito di scegliere un metodo per la *verifica_G* e *validazione_G* del materiale realizzato dal team. Il documento deve contenere:

- **Metodo di verifica:** vengono stabilite le procedure di controllo sulla qualità di processo e di prodotto.
- **Metriche:** è necessario stabilire delle metriche oggettive per i documenti e i processi software;
- **Gestione della revisione:** si devono stabilire delle modalità per comunicare le anomalie e le procedure per controllare la qualità di processo;
- **Pianificazione collaudo:** è necessario definire dettagliatamente le metodologie di collaudo del progetto;
- **Resoconto attività verifica:** alla fine di ogni attività svolta si devono riportare le metriche relative e un resoconto sulla verifica effettuata.

2.2 Sviluppo

2.2.1 Scopo

Questa sezione affronta le attività e i compiti svolti dal gruppo al fine di produrre il software richiesto dall'azienda IKS Srl.

2.2.2 Aspettative

Al fine di implementare correttamente questa attività vi sono le seguenti aspettative:

- Realizzare un prodotto software conforme alle richieste del proponente
- Realizzare un prodotto software che soddisfi i test di verifica
- Realizzare un prodotto software che soddisfi i test di validazione
- Fissare degli obiettivi di *sviluppo*_G
- Fissare eventuali vincoli tecnologici
- Fissare i vincoli di design

2.2.3 Descrizione

Il processo di sviluppo si svolge in accordo con lo standard ISO/IEC 12207. Pertanto si compone delle seguenti attività:

- Analisi dei requisiti
- Progettazione
- Codifica

2.2.4 Attività

2.2.4.1 Analisi dei requisiti

2.2.4.1.1 Scopo Lo scopo di questa analisi è quella di elencare nel modo più preciso possibile, tutti i requisiti del progetto. I requisiti sono estrapolati da diverse fonti:

- Specifica del capitolato
- Incontri tenuti direttamente con il proponente
- Casi d'uso

Dopo aver svolto questo processo viene stilato in modo formale un documento di analisi chiamato *Analisi dei Requisiti v1.0.0*. Quest'ultimo è steso dagli *Analisti* e contiene una lista dei requisiti e dei casi d'uso. In questo modo sarà possibile effettuare e definire dei test di superamento del software.

2.2.4.1.2 Aspettative L'obiettivo di questa attività è quello di definire in modo formale e dettagliato tutti i requisiti richiesti dal proponente.

2.2.4.1.3 Descrizione Nel documento inerente a quest'analisi sono specificati i requisiti analizzati. Il metodo utilizzato per l'analisi dei requisiti è quella dei casi d'uso.

2.2.4.1.4 Casi d'uso

2.2.4.1.5 Codice identificativo

2.2.4.1.6 Requisiti

2.2.4.1.7 Codice identificativo

2.2.4.1.8 UML I diagrammi UML devono essere realizzati usando la versione del linguaggio *v2.0*

2.2.4.2 Progettazione

2.2.4.2.1 Scopo

2.2.4.2.2 Aspettative

2.2.4.2.3 Descrizione

2.2.4.2.4 Specifica Tecnica

2.2.4.2.5 Definizione di Prodotto

2.2.4.3 Codifica

2.2.4.3.1 Scopo Questa attività ha come scopo l'effettiva implementazione del prodotto software richiesto. In questa fase dunque si concretizzano attraverso la codifica le funzionalità previste dai requisiti concordati.

2.2.4.3.2 Aspettative Obiettivo dell'attività è la creazione di un prodotto software conforme alle richieste del proponente

2.2.4.3.3 Descrizione L'attività deve rispettare quando imposto dai documenti *Definizione di prodotto* e *Piano di Qualifica*

2.2.4.3.4 Stile di codifica Al fine di garantire uniformità all'intera *codebase_G*, ciascun membro del gruppo è obbligato a rispettare le seguenti norme:

- Formattazione: è richiesto l'uso di uno spazio (" ") ove possibile per rendere il codice di facile comprensione. Di seguito alcuni esempi di buona e cattiva formattazione.

```
1  int a=3; // BAD
2  int a = 3; // GOOD
3  int a = 3,b = 5; // BAD
4  int a = 3, b = 4; // GOOD
5
6  getFoo(a,b,c,d) // BAD
7  getFoo(a, b, c, d) // GOOD
8
9  if (a==5){ // BAD
10 if (a == 5) { // GOOD
```

- Indentazione: sono richiesti 4 spazi (si consiglia di impostare adeguatamente il proprio *IDE_G*), inoltre non sono permessi spazi bianchi o tabulazioni a fine riga.
- Lunghezza linee di codice: una riga di codice non deve superare i 100 caratteri, in caso contrario spezzare la riga di codice andando a capo

- Nomi: i nomi delle funzioni e delle variabili devono essere significativi e devono cominciare con la lettera minuscola e seguire la notazione *camelCase*. I nomi delle classi devono avere la prima lettera maiuscola mentre i nomi delle costanti devono essere scritti interamente in maiuscolo.

```
1 var foo; // BAD
2 var profiledata // BAD
3 var profileData // GOOD
```

- Lingua: i nomi delle variabili i metodi le funzioni e i commenti vanno scritti in inglese.
- Lunghezza metodi e funzioni: i corpo dei metodi e delle funzioni non deve superare le 40 linee e i due gradi di indentazione. Superare tali limiti è chiaro segno che la funzione / metodo necessita di essere spezzata. A volte però può risultare preferibile mantenere tutta la logica su un solo metodo per facilitarne la comprensibilità.
- Strutture di controllo (if, switch, for, etc...): le parentesi graffe per i costrutti che le prevedono devono essere in linea con il costrutto stesso. E' prevista l'omissione delle graffe soltanto quando il corpo della struttura è di una sola riga.

```
1 if (...) { // GOOD
2     // CODICE
3 } else {
4     // CODICE
5 }
6
7
8 if (...) // BAD
9 {
10     // CODE
11 }
12 else
13 {
14     // CODE
15 }
```

- Commenti: i commenti devono essere esaustivi e non prolissi in modo da aiutare il lettore a comprendere al meglio ciò che si sta leggendo.

E' inoltre consentito l'utilizzo di particolari tipi di commento con lo scopo di aiutare la stesura del codice. Vi sono:

- TODO: soluzione temporanee, memento di ogni tipo o zone che necessitano di un'implementazione.

```
1 // TODO: this solution stinks
2 // TODO: Write that query
```

- HOW: per segnalare che non si ha capito l'implementazione o il funzionamento di una porzione di codice e che si necessita di uno studio più approfondito

```
1 // HOW: need to study this API call , I m puzzled
```

- FIX: quando una particolare implementazione necessita di essere riparata o sistemata.

```
1 // FIX: this implementation doesn t work with IE
```

Questi commenti vanno scritti tutti in maiuscolo e seguiti da due punti (:), questa sintassi favorisce la loro individuazione da particolari *tool* che ne permettono una consultazione agevolata.

2.2.4.3.5 Intestazione Ogni *file_G* contenente codice sorgente deve avere la seguente intestazione:

```
1      /*
2
3      * File : nome file
4      * Version : versione file
5      * Type : tipo file
6      * Date : data di creazione
7      * Author : nome autore / i
8      * E - mail : email autore / i
9      *
10     * License : tipo licenza
11     *
12     * Avvertenze : lista avvertenze e limitazioni
13     *
14     * Descrizione: breve descrizione del contenuto del file
15     *
16     * Registro modifiche :
17     * Autore || Data || breve descrizione modifiche
18     *
19     */
```

2.2.4.3.6 Versionamento

2.2.4.3.7 Ricorsione E' caldamente sconsigliato l'utilizzo di ricorsione, esistono però dei casi in cui la soluzione ricorsiva rende il problema triviale, in tal caso è richiesta la verifica di correttezza e di terminazione della ricorsione.

2.2.5 Strumenti

2.2.5.1 Trender

2.2.5.2 Astah

2.2.5.3 IntelliJ IDEA

3 Processi Organizzativi

3.1 Gestione

3.1.1 Scopo

3.1.2 Aspettative

3.1.3 Descrizione

3.1.4 Ruoli di progetto

3.1.4.1 Amministratore di Progetto

3.1.4.2 Responsabile di Progetto

3.1.4.3 Analista

3.1.4.4 Progettista

3.1.4.5 Verificatore

3.1.4.6 Programmatore

3.1.5 Procedure

3.1.5.1 Gestione delle comunicazioni

3.1.5.1.1 Comunicazioni interne

3.1.5.1.2 Comunicazioni esterne

3.1.5.2 Gestione degli incontri

3.1.5.2.1 Incontri interni

3.1.5.2.2 Incontri esterni

3.1.5.3 Gestione degli strumenti di coordinamento

3.1.5.3.1 Ticketing

3.1.5.4 Gestione degli strumenti di versionamento

3.1.5.4.1 Repository

3.1.5.4.2 Struttura del repository

3.1.5.4.3 Tipi di file e .gitignore

3.1.5.4.4 Norme sui commit

3.1.5.5 Gestione dei rischi

3.1.6 Strumenti

3.1.6.1 Sistema Operativo

3.1.6.2 Slack

3.1.6.3 Telegram

3.1.6.4 Wrike

3.1.6.5 Git

3.1.6.6 Github

4 Processi di Supporto

4.1 Documentazione

4.1.1 Scopo

4.1.2 Aspettative

4.1.3 Descrizione

4.1.4 Procedure

4.1.4.1 Approvazione dei documenti

4.1.5 Template

4.1.6 Struttura dei documenti

4.1.6.1 Prima pagina

4.1.6.2 Registro delle modifiche

4.1.6.3 Indice

4.1.6.4 Contenuto principale

4.1.6.5 Note a piè di pagina

4.1.7 Versionamento

4.1.8 Norme tipografiche

4.1.8.1 Stile del testo

4.1.8.2 Elenchi puntati

4.1.8.3 Formati comuni

4.1.8.4 Sigle

4.1.9 Elementi grafici

4.1.9.1 Tabelle

4.1.9.2 Immagini

4.1.10 Classificazione dei documenti

4.1.10.1 Documenti informali

4.1.10.2 Documenti formali

4.1.10.3 Verbali

4.1.11 Strumenti

4.1.11.1 L^AT_EX

4.1.11.2 TexStudio

4.1.11.3 Lucidchart

4.2 Verifica

4.2.1 Scopo

4.2.2 Aspettative

4.2.3 Descrizione

4.2.4 Attività

4.2.4.1 Analisi

4.2.4.1.1 Analisi Statica

4.2.4.1.2 Analisi dinamica

4.2.4.2 Test

4.2.4.2.1 Test di unità

4.2.4.2.2 Test di integrazione

4.2.4.2.3 Test di sistema

4.2.4.2.4 Test di regressione

4.2.4.2.5 Test di accettazione

4.2.5 Strumenti

4.2.5.1 Verifica ortografica

4.2.5.2 Validazione W3C

4.2.5.3 Analisi statica

4.2.5.4 Analisi dinamica

4.2.5.5 Metriche