



Titolo del documento

SWEefty - 4 Dicembre 2017

Informazioni sul documento

Versione	0.0
Redazione	Alberto Galinaro
	Davide Zago
	Elia Montecchio
	Francesco Parolini
	Giuseppe Merlino
Verifica	Lisa Parma
	Paolo Eccher
	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

Indice

1	Introduzione	5
1.1	Scopo del documento	5
1.2	Scopo del prodotto	5
1.3	Ambiguità	5
1.4	Riferimenti	5
1.4.1	Normativi	5
1.4.2	Informatici	5
2	Processi primari	5
2.1	Fornitura	5
2.1.1	Scopo	5
2.1.2	Aspettative	5
2.1.3	Descrizione	5
2.1.4	Attività	5
2.1.4.1	Studio di fattibilità	5
2.1.4.2	Piano di Progetto	5
2.1.4.3	Piano di Qualifica	5
2.2	Sviluppo	6
2.2.1	Scopo	6
2.2.2	Aspettative	6
2.2.3	Descrizione	6
2.2.4	Attività	6
2.2.4.1	Analisi dei requisiti	6
2.2.4.1.1	Scopo	6
2.2.4.1.2	Aspettative	6
2.2.4.1.3	Descrizione	6
2.2.4.1.4	Casi d'uso	6
2.2.4.1.5	Codice identificativo	6
2.2.4.1.6	Requisiti	6
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	7
2.2.4.3.3	Descrizione	7
2.2.4.3.4	Stile di codifica	7
2.2.4.3.5	Intestazione	7
2.2.4.3.6	Versionamento	7
2.2.4.3.7	Ricorsione	7
2.2.5	Strumenti	7
2.2.5.1	Trender	7
2.2.5.2	Astah	7
2.2.5.3	IntelliJ IDEA	7
3	Processi Organizzativi	7
3.1	Gestione	7
3.1.1	Scopo	7
3.1.2	Aspettative	7

3.1.3	Descrizione	7
3.1.4	Ruoli di progetto	7
3.1.4.1	Amministratore di Progetto	7
3.1.4.2	Responsabile di Progetto	7
3.1.4.3	Analista	7
3.1.4.4	Progettista	7
3.1.4.5	Verificatore	7
3.1.4.6	Programmatore	7
3.1.5	Procedure	8
3.1.5.1	Gestione delle comunicazioni	8
3.1.5.1.1	Comunicazioni interne	8
3.1.5.1.2	Comunicazioni esterne	8
3.1.5.2	Gestione degli incontri	8
3.1.5.2.1	Incontri interni	8
3.1.5.2.2	Incontri esterni	8
3.1.5.3	Gestione degli strumenti di coordinamento	8
3.1.5.3.1	Ticketing	8
3.1.5.4	Gestione degli strumenti di versionamento	8
3.1.5.4.1	Repository	8
3.1.5.4.2	Struttura del repository	8
3.1.5.4.3	Tipi di file e .gitignore	8
3.1.5.4.4	Norme sui commit	8
3.1.5.5	Gestione dei rischi	8
3.1.6	Strumenti	8
3.1.6.1	Sistema Operativo	8
3.1.6.2	Slack	8
3.1.6.3	Telegram	8
3.1.6.4	Wrike	8
3.1.6.5	Git	8
3.1.6.6	Github	8
4	Processi di Supporto	9
4.1	Documentazione	9
4.1.1	Scopo	9
4.1.2	Aspettative	9
4.1.3	Descrizione	9
4.1.4	Procedure	9
4.1.4.1	Approvazione dei documenti	9
4.1.5	Template	9
4.1.6	Struttura dei documenti	9
4.1.6.1	Prima pagina	9
4.1.6.2	Registro delle modifiche	10
4.1.6.3	Indici	10
4.1.6.4	Formattazione generale delle pagine	10
4.1.6.4.1	Intestazione	10
4.1.6.4.2	Piè di pagina	10
4.1.6.5	Note a piè di pagina	11
4.1.7	Versionamento	11
4.1.8	Norme tipografiche	11
4.1.8.1	Stile del testo	11
4.1.8.2	Elenchi puntati	12
4.1.8.3	Formati comuni	12
4.1.8.4	Sigle	12
4.1.9	Elementi grafici	13
4.1.9.1	Tabelle	13
4.1.9.2	Immagini	13

4.1.10	Classificazione dei documenti	13
4.1.10.1	Documenti informali	13
4.1.10.2	Documenti formali	13
4.1.10.3	Verbali	13
4.1.11	Strumenti	13
4.1.11.1	L ^A T _E X	13
4.1.11.2	TexStudio	13
4.1.11.3	Lucidchart	13
4.2	Verifica	13
4.2.1	Scopo	13
4.2.2	Aspettative	13
4.2.3	Descrizione	13
4.2.4	Attività	13
4.2.4.1	Analisi	13
4.2.4.1.1	Analisi Statica	13
4.2.4.1.2	Analisi dinamica	14
4.2.4.2	Test	14
4.2.4.2.1	Test di unità	14
4.2.4.2.2	Test di integrazione	14
4.2.4.2.3	Test di sistema	14
4.2.4.2.4	Test di regressione	14
4.2.4.2.5	Test di accettazione	14
4.2.5	Strumenti	14
4.2.5.1	Verifica ortografica	14
4.2.5.2	Analisi statica	14
4.2.5.3	Analisi dinamica	15
4.2.5.4	Metriche	15

1 Introduzione

1.1 Scopo del documento

1.2 Scopo del prodotto

1.3 Ambiguità

1.4 Riferimenti

1.4.1 Normativi

1.4.2 Informatici

2 Processi primari

2.1 Fornitura

2.1.1 Scopo

Lo scopo di questo *processo_Gs* (aggiungere pedice G ?) e' 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 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:

- Determinare aspetti chiave per soddisfare i bisogni del proponente
- determinare i vincoli sui processi e sui requisiti
- stimare i costi
- concordare la qualifica del prodotto

2.1.3 Descrizione

Il gruppo intende mantenere un costante dialogo con il proponente in modo da poter avere un riscontro sull'efficacia del lavoro svolto e sull'applicazione delle tecnologie coinvolte

2.1.4 Attività

2.1.4.1 Studio di fattibilità

2.1.4.2 Piano di Progetto

2.1.4.3 Piano di Qualifica

2.2 Sviluppo

2.2.1 Scopo

2.2.2 Aspettative

2.2.3 Descrizione

2.2.4 Attività

2.2.4.1 Analisi dei requisiti

2.2.4.1.1 Scopo

2.2.4.1.2 Aspettative

2.2.4.1.3 Descrizione

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

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

2.2.4.3.2 Aspettative

2.2.4.3.3 Descrizione

2.2.4.3.4 Stile di codifica

2.2.4.3.5 Intestazione

2.2.4.3.6 Versionamento

2.2.4.3.7 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

Questo processo include e descrive le modalità di redazione e manutenzione dei documenti e le convenzioni adottate per la scrittura di questi, durante il ciclo di vita del prodotto software.

4.1.2 Aspettative

I risultati che ci aspettiamo di ottenere da una corretta implementazione di questo processo sono:

- una concezione attenta ed univoca dei documenti che vanno redatti durante il ciclo di vita del software;
- la scrittura di documenti precisi, coerenti e conformi a delle regole;
- l'individuazione di una collezione di norme e convenzioni per la redazione di documentazione coerente e valida.

4.1.3 Descrizione

In questa sezione devono essere indicate tutte le norme e le convenzioni adottate dal gruppo, per consentire la stesura di una documentazione valida e coerente.

4.1.4 Procedure

Il gruppo si è servito del linguaggio di markup \LaTeX per la stesura della documentazione.

4.1.4.1 Approvazione dei documenti

Ogni documento non formale in corrispondenza del completamento della sua stesura dovrà essere sottoposto al *Responsabile di Progetto*, che dovrà delegare ai *Verificatori* il controllo del contenuto e della forma. Nel caso gli anzidetti *Verificatori* rilevino degli errori, sarà loro compito notificarli al *Responsabile di Progetto*, che a sua volta affiderà il redattore del documento il compito di correggerli. Questo ciclo va eseguito fino a quando il documento non viene considerato completamente corretto dai *Verificatori*. In caso di assenso sulla correttezza e sulla qualità il documento può essere reputato un documento formale. In caso contrario spetta al *Responsabile di Progetto* comunicare le ragioni per cui il documento non è stato giudicato corretto, esplicitando le modifiche da apportare.

4.1.5 Template

Per agevolare la redazione della documentazione è stato creato un template \LaTeX contenente tutte le impostazioni stilistiche e grafiche citate in questo documento.

4.1.6 Struttura dei documenti

4.1.6.1 Prima pagina

Ogni documento è caratterizzato da una prima pagina che contiene le seguenti informazioni sul documento:

- Logo del gruppo;
- Titolo del documento;
- Nome del gruppo;

- Nome del progetto;
- Versione del documento;
- Cognome e nome dei redattori del documento;
- Cognome e nome dei verificatori del documento;
- Cognome e nome del responsabile approvatore del documento;
- Destinazione d'uso del documento;
- Lista di distribuzione del documento;
- Una breve descrizione del documento.

4.1.6.2 Registro delle modifiche

La seconda pagina di ogni documento contiene il diario delle modifiche del documento. Ogni riga del diario delle modifiche contiene:

- Un breve sommario delle modifiche svolte;
- Cognome e nome dell'autore;
- Ruolo dell'autore;
- Data della modifica;
- Versione del documento dopo la modifica.

La tabella contenente le modifiche è ordinata per data in ordine decrescente, affinché la prima riga contenga la versione attuale del documento.

4.1.6.3 Indici

In ogni documento, esclusi i verbali, è presente un indice delle sezioni, un indice delle figure e un indice delle tabelle. Nel caso non siano presenti figure o tabelle i rispettivi indici verranno omessi. La notazione numerica di ogni indice deve partire da 1 e le sottosezioni devono essere separate da un punto. Anche la notazione della sottosezione parte da 1.

4.1.6.4 Formattazione generale delle pagine

Il template prevede dei margini orizzontali e verticali che devono essere rispettati in ogni pagina. Ad esclusione della prima, tutte le pagine contengono un'intestazione ed un piè di pagina.

4.1.6.4.1 Intestazione

L'intestazione è così strutturata:

- Logo del gruppo posto a sinistra;
- Indirizzo di posta elettronica del gruppo posto a destra;
- Ruolo dell'autore;

4.1.6.4.2 Piè di pagina

Il piè di pagina è così strutturato:

- Nome e versione del documento corrente, posti a sinistra;
- Numerazione progressiva della pagina rispetto al totale posta a destra.

4.1.6.5 Note a piè di pagina

In caso di presenza in una pagina interna di note da esplicitare, esse vanno indicate nella pagina corrente, in basso a sinistra. Ogni nota deve riportare un numero e una descrizione.

4.1.7 Versionamento

Tutti i documento, esclusi i verbali, devono essere versionati, affinché si possa in ogni momento conoscerne la storia. Ogni versione di un documento deve essere appuntata nel Registro delle Modifiche. Si deve adottare il seguente formato per registrare ogni modifica: **vX.Y.Z** dove:

- **X**
 - Parte da 0;
 - Viene incrementato dal *Responsabile di Progetto* in concomitanza con l'approvazione del documento;
 - Arriva fino al numero di revisioni a cui è sottoposto il documento.
- **Y**
 - Parte da 0;
 - Viene incrementato dal *Verificatore_G* in concomitanza di ogni verifica;
 - Quando X viene incrementato Y ritorna a 0.
- **Z**
 - Parte da 0;
 - Viene incrementato dal *Redattore* in concomitanza di ogni modifica;
 - Quando Y viene incrementato Z ritorna a 0.

4.1.8 Norme tipografiche

Per la redazione della documentazione bisogna attenersi a queste norme.

4.1.8.1 Stile del testo

- **Grassetto:** il grassetto può essere utilizzato nei seguenti casi:
 - **Elenchi puntati:** in questi casi può essere utilizzato il grassetto per evidenziare il concetto sviluppato nella continuazione del punto;
 - **Altri casi:** per evidenziare particolari passaggi o parole chiave.
- **Corsivo:** il corsivo deve essere utilizzato nei seguenti casi:
 - **Citazioni:** quando si deve citare una frase questa va scritta in corsivo;
 - **Abbreviazioni:** quando possibile si deve preferire una parola completa ad un'abbreviazione;
 - **Nomi particolari:** il corsivo deve essere utilizzato quando si parla di figure particolari (es. *Progettista*);
 - **Documenti:** il corsivo deve essere utilizzato quando si parla di documenti (es. *Glossario*);
 - **Altri casi:** in altre situazione, il corsivo va utilizzato per mettere in rilievo passaggi o parole significativi, evidenziare riferimenti ai documenti interni o esterni.
- **Maiuscolo:** l'utilizzo di parole completamente in maiuscolo è riservato solo agli acronimi.

4.1.8.2 Elenchi puntati

Ogni punto dell'elenco deve terminare con un punto e virgola, tranne l'ultimo che deve terminare con un punto. La prima parola deve avere la lettera maiuscola, a meno di casi particolari (es. nome di un file);

4.1.8.3 Formati comuni Per rappresentare le seguenti entità vanno usate le seguenti modalità:

- **Orari: HH:MM**

- **HH:** va da 0 a 23 e rappresenta l'ora;
- **MM:** va da 0 a 59 e rappresenta i minuti.

- **Date: AAAA-MM-GG**

- **AAAA:** rappresenta l'anno;
- **MM:** rappresenta il mese;
- **GG:** rappresenta il giorno.

- **Termini ricorrenti:**

- **Nomi proprio:** ogni nome proprio va scritto con il formalismo *Nome Cognome*;
- **Ruoli di progetto:** ogni ruolo va scritto con l'iniziale maiuscola e in corsivo;
- **Nomi dei documenti:** ogni documenti va scritto con l'iniziale maiuscola e in corsivo.

4.1.8.4 Sigle

È previsto l'utilizzo delle seguenti sigle:

- AR: *Analisi dei Requisiti*;
- PP: *Piano di Progetto*;
- NP: *Norme di Progetto*;
- SF: *Studio di Fattibilità*;
- PQ: *Piano di Qualifica*;
- ST: *Specifica Tecnica*;
- MU: *Manuale utente_G*;
- DP: *Definizione di Prodotto*;
- RR: Revisione dei requisiti;
- RP: Revisione di progettazione;
- RQ: Revisione di qualifica;
- RA: Revisione di accettazione;
- Re: *Responsabile di Progetto*;
- Am: *Amministratore di Progetto* (o Project manager?);
- An: *Analista*;
- Pt: *Progettista*;
- Pr: *Programmatore_G*;
- Ve: *Verificatore*.

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

L'analisi statica è una tecnica utilizzata per identificare errori all'interno di documenti e del codice sorgente, senza la necessità eseguirlo, è applicabile durante tutto il loro ciclo di vita in due diverse modalità:

- **Walkthrough:**

consiste nel leggere il documento/codice cercando anomalie senza avere un'idea chiara di che tipo di errori possono essere trovati, è un'attività onerosa e non efficiente ma necessaria durante le prime fasi di progetto dove sarà la principale forma di verifica adottata, in quanto non è chiaro fin dall'inizio che tipo di errori si possono fare. Questo metodo è utile per stilare una *checklist_G* dove verranno archiviati gli errori più comuni. Vista la sua scarsa efficacia normalmente viene effettuata da più persone.

- **Inspection:**

consiste nella lettura mirata del documento/codice per localizzare possibili errori specificati dalla lista di controllo. Ha un basso costo e diventa più efficace con l'esperienza e l'estensione della lista di controllo, per questo può essere effettuata da una persona sola.

Ogni errore rilevato va discusso con l'autore allo scopo di concordare una modifica.

4.2.4.1.2 Analisi dinamica

L'analisi dinamica è una forma di analisi del software che richiede l'esecuzione del codice sorgente. Viene effettuata tramite dei test che verificano il corretto funzionamento del prodotto, in caso di anomalie questi test aiutano ad identificare l'errore. I test devono essere ripetibili ovvero con lo stesso input e nello stesso ambiente deve essere in grado di ottenere sempre lo stesso output. Per ogni test sono quindi definiti i seguenti parametri:

- **Ambiente:** Sistema hardware e software sul quale si svolgerà il test.
- **Stato iniziale**
- **Input**
- **Output**
- **Istruzioni aggiuntive:** istruzioni su come va eseguito il test e su come vanno interpretati gli output

4.2.4.2 Test

4.2.4.2.1 Test di unità

I test di unità verificano che le singole *unita'_G* di prodotto software funzionino correttamente, le unità non possono passare al test di integrazione se non passano questo test. Comunemente per i test di unità vengono utilizzati *driver_G* e *stub_G* che simulano il chiamante e un'unità chiamata.

4.2.4.2.2 Test di integrazione

I test di integrazione sono il passo successivo ai test di unità servono a verificare che due o più unità, precedentemente verificate, funzionino correttamente una volta combinate. Questo tipo di test verifica la corretta collaborazione tra le varie unità e aiuta a trovare eventuali errori non rilevati nei test precedenti. L'obiettivo finale di questi test è arrivare a testare l'intero prodotto costituito accorpando tutte le singole unità.

4.2.4.2.3 Test di sistema

I test di sistema validano il prodotto software ovvero si accertano che rispettino i requisiti concordati, vengono effettuati quando si ritiene che il prodotto abbia raggiunto una versione definitiva.

4.2.4.2.4 Test di regressione

I test di regressione devono essere fatti ogni volta che viene fatta una modifica a una componente software, consistono nel rifare i test di unità e integrazione necessari per accertarsi che tale modifica non causi errori nella componente in cui è stata fatta o in parti del software collegate. Un buon incapsulamento limita il numero di test da effettuare in questa fase.

4.2.4.2.5 Test di accettazione

Consiste nel collaudo del software in presenza del proponente, se questo test ha esito positivo il prodotto può essere rilasciato.

4.2.5 Strumenti

4.2.5.1 Verifica ortografica Viene utilizzato **GNU Aspell**, uno spell checker open source in grado di gestire più dizionari contemporaneamente.

4.2.5.2 Analisi statica Per l'analisi statica del codice JavaScript viene utilizzato **SonarSource**, questo prodotto è in grado di rilevare bug e problemi di sicurezza, è inoltre disponibile su vari *IDE_G* come Eclipse e IntelliJ.

4.2.5.3 Analisi dinamica Viene utilizzato **Jest** come piattaforma per l'esecuzione di test su codice JavaScript, comodo per la sua semplicità di utilizzo.

4.2.5.4 Metriche per il calcolo delle metriche viene utilizzato **SonarQube**, software che permette di stabilire un grado di qualità che tutto il gruppo deve rispettare, inoltre permette di visualizzare grafici per controllare la qualità del progetto nel periodo di sviluppo.