



Titolo del documento

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

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	8
3.1.4.2	Analista	8
3.1.4.3	Progettista	8
3.1.4.4	Verificatore	8
3.1.4.5	Programmatore	8
3.1.4.6	Project Manager	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	10
3.1.5.3	Gestione degli strumenti di coordinamento	10
3.1.5.3.1	Ticketing	10
3.1.5.3.2	Struttura del workspace di comunicazioni interne	10
3.1.5.4	Gestione degli strumenti di versionamento	10
3.1.5.4.1	Repository	11
3.1.5.4.2	Struttura del repository	11
3.1.5.4.3	Norme sui nomi dei files	11
3.1.5.4.4	Files e .gitignore	12
3.1.5.4.5	Norme su commit	12
3.1.5.5	Gestione dei rischi	12
3.1.6	Strumenti	12
3.1.6.1	Sistema Operativo	12
3.1.6.2	Slack	12
3.1.6.3	Trello	12
3.1.6.4	Github	12
3.1.6.5	Git	13
3.1.6.6	Gitkraken	13
4	Processi di Supporto	13
4.1	Documentazione	13
4.1.1	Scopo	13
4.1.2	Aspettative	13
4.1.3	Descrizione	13
4.1.4	Procedure	13
4.1.4.1	Approvazione dei documenti	13
4.1.5	Template	13
4.1.6	Struttura dei documenti	13
4.1.6.1	Prima pagina	13
4.1.6.2	Registro delle modifiche	13
4.1.6.3	Indice	13
4.1.6.4	Contenuto principale	13
4.1.6.5	Note a piè di pagina	13
4.1.7	Versionamento	13
4.1.8	Norme tipografiche	13
4.1.8.1	Stile del testo	13
4.1.8.2	Elenchi puntati	13
4.1.8.3	Formati comuni	14
4.1.8.4	Sigle	14
4.1.9	Elementi grafici	14
4.1.9.1	Tabelle	14
4.1.9.2	Immagini	14

4.1.10	Classificazione dei documenti	14
4.1.10.1	Documenti informali	14
4.1.10.2	Documenti formali	14
4.1.10.3	Verbali	14
4.1.11	Strumenti	14
4.1.11.1	L ^A T _E X	14
4.1.11.2	TexStudio	14
4.1.11.3	Lucidchart	14
4.2	Verifica	14
4.2.1	Scopo	14
4.2.2	Aspettative	14
4.2.3	Descrizione	14
4.2.4	Attività	14
4.2.4.1	Analisi	14
4.2.4.1.1	Analisi Statica	14
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	15
4.2.4.2.3	Test di sistema	15
4.2.4.2.4	Test di regressione	15
4.2.4.2.5	Test di accettazione	15
4.2.5	Strumenti	15
4.2.5.1	Verifica ortografica	15
4.2.5.2	Validazione W3C	15
4.2.5.3	Analisi statica	15
4.2.5.4	Analisi dinamica	15
4.2.5.5	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

Questo *processo_G* ha come scopo la redazione del documento *Piano di Progetto*.*forse*

3.1.2 Aspettative

Ci si aspetta di *non ne ho idea, servono altri esempi*

3.1.3 Descrizione

non ne ho idea, servono altri esempi

3.1.4 Ruoli di progetto

I ruoli di progetto rappresentano le figure professionali che lavoreranno al progetto. Ogni membro dovrà ricoprire ciascun ruolo almeno una volta. Come ciò sarà pianificato e verificato verrà specificato nel documento *Piano di Progetto v1.0.0*. I ruoli sono:

3.1.4.1 Amministratore di Progetto

L'*Amministratore di Progetto* è la figura professionale che si deve occupare di gestire l'ambiente di lavoro del gruppo. Nello specifico deve

- Ricercare nuovi strumenti che migliorino la produttività del gruppo
- Imparare ad utilizzare adeguatamente gli strumenti utilizzati
- Se necessario spiegare agli altri componenti del gruppo come utilizzare correttamente tali strumenti
- Occuparsi del controllo di versione del prodotto
- Occuparsi della configurazione del prodotto

è presente per tutta la durata del progetto.

3.1.4.2 Analista

L'*Analista* è la figura professionale che si occupa dell'analisi dei requisiti del prodotto e del dominio applicativo. Deve

- Analizzare i requisiti del prodotto. Per fare ciò dovrà parlare in prima persona con il committente
- Analizzare i requisiti di dominio
- Redigere il documento *Studio di Fattibilità*
- Redigere il documento *Analisi dei Requisiti*

è presente solamente nella fase iniziale del progetto.

3.1.4.3 Progettista

Il *Progettista* è la figura professionale che si occupa della progettazione del prodotto. Deve

- Comprendere a fondo i requisiti nel documento *Analisi dei Requisiti*
- Progettare un'*architettura_G* per il prodotto
- Scegliere le tecnologie che si utilizzeranno per realizzare il prodotto, in modo tale che essere permettano di soddisfare i requisiti
- Redigere il documento *Specifica Tecnica*
- Redigere il documento *Definizione di prodotto* **ATTENZIONE: questi due documenti esistono ancora?**

è presente nella fase di progettazione del prodotto software e *può* seguire il progetto fino al termine.

3.1.4.4 Verificatore

Il *Verificatore* è la figura professionale che si occupa dell'attività di verifica. Deve

- Verificare che ciascuna attività svolta sia conforme alle norme stabilite nel progetto
- Redigere il documento *Piano di Qualifica*

è presente per l'intera durata del progetto.

3.1.4.5 Programmatore

Il *Programmatore* è la figura professionale che si occupa della codifica del prodotto. Deve

- Scrivere il codice del prodotto software che rispetti le decisioni del *Progettista*
- Redigere il documento *Manuale Utente*

è presente durante la fase di codifica del prodotto.

3.1.4.6 Project Manager

Il *Project Manager* è la figura professionale funge da punto di riferimento per il gruppo, il fornitore e il *committente_G* in merito al progetto che amministra. Deve

- Organizzare incontri interni ed esterni
- Pianificare le attività svolte dal gruppo, suddividendole in compiti
- Individuare per ciascun compito un membro del gruppo per svolgerlo
- Analizzare, monitorare e gestire i rischi

è presente per l'intera durata del progetto.

3.1.5 Procedure

3.1.5.1 Gestione delle comunicazioni

3.1.5.1.1 Comunicazioni interne

Le comunicazioni interne sono gestite con la piattaforma *Slack_G*. Essa rende possibile creare un *workspace_G* comune al gruppo suddiviso in più *canali di comunicazione_G*, ad ognuno dei quali sono associati uno scopo ed un sottoinsieme dei membri del gruppo. *Slack* permette anche a due componenti del gruppo di mandarsi messaggi privati. Il maggiore vantaggio rispetto all'utilizzo di un'applicazione di messaggistica come *Telegram_G* o *Whatsapp_G* sta nel poter creare uno spazio di comunicazione condiviso che permetta di gestire più categorie di comunicazioni in un singolo *workspace*.

3.1.5.1.2 Comunicazioni esterne

La gestione delle comunicazioni esterne è compito del *Project Manager*. Egli deve utilizzare l'indirizzo di posta elettronica:

sweeftyteam@gmail.com

Il *Project manager* se lo riterrà opportuno potrà informare i membri del gruppo delle eventuali comunicazioni esterne tramite il canale Slack *email*.

3.1.5.2 Gestione degli incontri

3.1.5.2.1 Incontri interni

Gli incontri interni del gruppo devono essere organizzati dal *Project Manager*. Egli deve:

1. Proporre tramite il canale Slack *incontri* una data, un'ora, un luogo e una motivazione per organizzare l'incontro. La data in cui ciò avviene deve essere almeno a due giorni di distanza dalla data proposta per incontrarsi. Quest'ultima regola può essere ignorata solamente nel caso in cui fosse strettamente necessaria una riunione tempestiva.
2. Raccogliere tramite il *Bot_G* Slack *Simple Poll_G* le risposte di partecipazione o assenza dei membri. Esse devono essere date entro e non oltre 24 ore di tempo dall'apertura del sondaggio.
3. Se almeno cinque membri avranno dato conferma di presenza il *Project Manager* confermerà che l'incontro si svolgerà con i parametri definiti nel punto 1. Se le conferme di presenza saranno tre o quattro starà al *Project Manager* decidere fra:
 - Confermare l'incontro con solo coloro che hanno confermato al presenza
 - Tornare al punto 1 cambiando i parametri data e/o ora ed eventualmente luogo

Tale decisione dovrà prendere in considerazione i benefici che porterebbe organizzare un incontro con tre o quattro membri e gli svantaggi che si avrebbero posticipando l'incontro. Se le conferme saranno meno di tre sarà necessario tornare al punto 1 cambiando i parametri data e/o ora ed eventualmente luogo

4. Nel caso in cui l'incontro venga confermato, stilare in un file nella cartella della repository *Docs/Verbali* l'ordine del giorno. Il file deve avere come nome la data in cui si terrà l'incontro, scritta nel formato indicato in **?? dove sta sta data?**. Nello stesso file dovrà essere verbalizzato ciò di cui si è discusso durante l'incontro nelle modalità descritte alla fine di questo paragrafo

Tutti i membri del team possono chiedere di organizzare un incontro al *Project Manager*, specificando una motivazione. Se quest'ultima verrà ritenuta sufficiente per organizzare un incontro si procederà ad avviare la procedura sopra descritta.

Inoltre, ad ogni incontro, un membro del gruppo verrà scelto dal *Project Manager* per stilare il verbale. Egli dovrà inserire nello stesso file creato dal *Project Manager* contenente l'ordine del giorno un riassunto scritto di ciò di cui si è discusso e ciò che è stato deciso.

3.1.5.2.2 Incontri esterni

Gli incontri esterni dovranno essere organizzati dal *Project Manager*. Sarà suo compito mettersi in contatto con il committente o il proponente tramite email e determinare data, ora e luogo dell'incontro. Quando ciò sarà definito dovrà avvisare i membri del gruppo sul canale Slack *incontri*. Anche per gli incontri esterni ciascun membro del gruppo può richiedere che venga organizzato un incontro con il committente o il proponente, specificandone la motivazione. Se essa sarà giudicata adeguata dal *Project Manager* egli dovrà procedere ad organizzare tale incontro. Sarà anche suo compito incaricare uno dei presenti di redigere un verbale scritto.

3.1.5.3 Gestione degli strumenti di coordinamento

3.1.5.3.1 Ticketing

Per la gestione del sistema di ticketing si utilizza la piattaforma *TrelloG*. Essa permette di definire un insieme di *tasksG*, ad ognuno dei quali viene assegnato un nome, una breve descrizione, un membro che sia incaricato di portarlo a termine, un *Verificatore*, una data di scadenza ed uno stato. Il *Project Manager* dovrà creare i tasks ed assegnare a ciascuno di essi un incaricato ed un *Verificatore*, specificando nella descrizione chi dei due svolge quale compito. Lo stato è rappresentato da una checklist composta da due punti: "completato" e "verificato". Se nessuna delle due opzioni è stata spuntata ciò significa che il task non è stato completato. Se solamente l'opzione "completato" è spuntata ciò significa che colui che era incaricato di portare a termine tale compito l'ha fatto. Se entrambe le opzioni sono spuntate significa che il task è anche stato verificato dal *Verificatore*. Non è possibile che sia spuntata l'opzione "verificato" senza che sia spuntata quella "completato". È compito dell'incaricato di portare a termine il task spuntare "completato" e del *Verificatore* spuntare "verificato".

3.1.5.3.2 Struttura del workspace di comunicazioni interne

Come detto in 3.1.5.1.1 lo strumento utilizzato per le comunicazioni interne è Trello. Il workspace del gruppo è organizzato nei seguenti canali:

- **#general** è il canale nel quale il gruppo si scambia informazioni per l'appunto di carattere generale, per esempio domande su come utilizzare una *featureG* di un determinato strumento piuttosto che avvisare gli altri di essere in anticipo per un incontro
- **#incontri** ha lo scopo di organizzare gli incontri interni al gruppo. Qui verranno effettuati i sondaggi di partecipazione tramite il bot *Simple Poll* e successivamente sarà confermato o meno l'incontro dal *Project Manager*
- **#todo** ha lo scopo di informare gli altri membri del gruppo che una certa porzione di un determinato task non è completa e potrà essere completata solo dopo che saranno state determinate delle norme nel gruppo. È compito di chi lascia il task incompleto informare gli altri tramite questo canale. I messaggi devono seguire la forma:

[*TODO*] spiegazione del problema

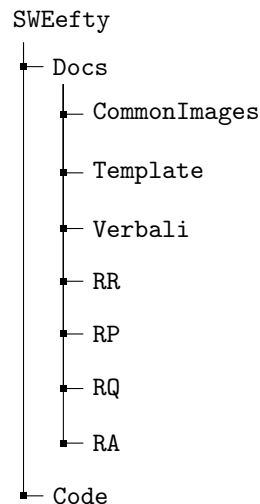
3.1.5.4 Gestione degli strumenti di versionamento

3.1.5.4.1 Repository

La piattaforma di *hosting_G* scelta per il repository è *GitHub_G*. Si utilizza una licenza "educational", che permette di avere fino a cinque repository private condivise. Di queste solamente una verrà effettivamente utilizzata. Sarà organizzata come descritto in 3.1.5.4.2

3.1.5.4.2 Struttura del repository

La struttura è la seguente



dove:

- **SWEefty** rappresenta la radice dello spazio di lavoro
- **Docs** è la directory in cui verranno inseriti tutti i documenti formali
- **Code** è la directory dove verrà inserita l'implementazione del codice
- **CommonImages** contiene le immagini comuni a tutti i documenti, per esempio il logo del gruppo
- **Template** contiene i files che compongono la struttura comune a tutti i documenti che verranno redatti
- **Verbali** contiene i files dei verbali degli incontri interni. Il nome di ogni file è identificato dalla data in cui è stata fatta la riunione del gruppo nel formato descritto in ?? **dove sta sto formato data?**
- **RR** contiene tutti i documenti da consegnare per la Revisione dei Requisiti
- **RP** contiene tutti i documenti da consegnare per la Revisione di Progettazione
- **RQ** contiene tutti i documenti da consegnare per la Revisione di Qualifica
- **RA** contiene tutti i documenti da consegnare per la Revisione di Accettazione

3.1.5.4.3 Norme sui nomi dei files

Ciascun file dovrà essere denominato seguendo la codifica:

NomeDelFile.ext

Dove NomeDelFile rappresenta il nome del file e ciascuna parola che compone tale nome è identificata da una lettera maiuscola. Ext rappresenta l'estensione del file. Nel caso in cui dovrebbero essere presenti lettere accentate nel nome, l'accento dovrà essere ignorato e sarà presente semplicemente la lettera non accettata. Per esempio il file sorgente .tex del documento *Studio di Fattibilità* verrà denominato **StudioDiFattibilita.tex**. **Ci mettiamo il controllo di versione nel nome? io non lo farei..forse**

3.1.5.4.4 Files e .gitignore

I file che si troveranno effettivamente nella directory `Docs` saranno solamente immagini `.jpg` e `.png`, file sorgente `LATEX` `.tex` e file `.pdf`. Tutti i prodotti intermedi del programma per la compilazione di file `.tex` `pdflatex` `.aux`, `.dvi`, `.fls`, `.log`, `.out`, `.toc` verranno inseriti nel file `.gitignore` che renderà la loro presenza trasparente a Git.

3.1.5.4.5 Norme su commit

Ogni volta che viene effettuata una modifica significativa ad un file oppure una serie di modifiche strettamente correlate su più files, va utilizzato il comando `commit` di `git`. In esso si deve specificare un messaggio che descriva brevemente ciò che è stato fatto. I caratteri di questo messaggio non devono essere più di 80. Deve seguire la semantica

[sigla file] breve descrizione

Per esempio l'aggiunta della sezione "XYZ" al file "NormeDiProgetto.tex" comporterebbe scrivere come messaggio informativo "[NDP] Aggiunta sezione XYZ". Prima di eseguire una `commit` va aggiornato il diario delle modifiche come descritto in ?? **dove sto sto diario?**

3.1.5.5 Gestione dei rischi

La gestione dei rischi è compito del *Project Manager*. Egli li deve identificare, inserirli nel *Piano di Progetto*_G e valutare una strategia per affrontarli. Nel caso in cui si presentassero nuovi rischi non osservati sarà sempre compito del *Project Manager* inserirli nel *Piano di Progetto*. Durante tutta la durata del *progetto*_G dovrà inoltre monitorare tali rischi e, solamente se necessario, ridefinire le strategie di progetto.

3.1.6 Strumenti

3.1.6.1 Sistema Operativo

I sistemi operativi utilizzati dai membri del gruppo sono:

- Ubuntu GNOME 17.10 x64
- Ubuntu 17.10 x64
- Windows 10 Home x64
- MacOS High Sierra x64

3.1.6.2 Slack

Slack è lo strumento adottato per gestire le comunicazioni interne al gruppo. Esso è gratuito e permette di creare in un workspace comune con più canali di comunicazione, contrassegnati da un cancelletto. Permette anche l'integrazione con altri strumenti utilizzati, quali *Drobbox*_G, Trello e GitHub. Queste *major features*_G lo rendono preferibile ad altre applicazioni di messaggistica quali Telegram o Whatsapp, più idonee ad un uso personale.

3.1.6.3 Trello

Trello è lo strumento adottato per gestire il sistema di ticketing. Permette l'utilizzo di estensioni, chiamate *powerup*_G, ad esempio integrazione con Slack o un calendario che mostri graficamente le scadenze per portare a termine certi tasks. La versione utilizzata è quella gratuita. La differenza con quella a pagamento consiste nell'avere a disposizione un singolo *powerup*.

3.1.6.4 Github

Github è la piattaforma scelta per l'hosting della repository. Essa si basa sul sistema di versionamento *git*_G, descritto in 3.1.6.5. Ha piani di sottoscrizioni gratuiti o a pagamento. La differenza sta nella possibilità di disporre di repository private. Con l'edizione "educational" da noi utilizzata si possono ottenere gratuitamente fino a cinque repository gratuite. GitHub offre inoltre un gran numero di tools utili per lo sviluppo aggiuntivi, come la gestione tramite interfaccia web delle *issues*_G.

3.1.6.5 Git

Git è il sistema per il controllo del versionamento utilizzato. Ha un'architettura distribuita ed è l'unico sistema di controllo di versione supportato da GitHub. Rispetto ad altri strumenti di controllo di versionamento, come *Mercurial_G*, ha un numero molto maggiore di comandi, il che rende difficile per un neofita l'utilizzo da riga di comando di git.

3.1.6.6 Gitkraken

Gitkraken_G è un'applicazione desktop *multiplatforma_G* che semplifica l'utilizzo di git, esponendo una chiara interfaccia grafica che si sostituisce all'utilizzo di git da riga di comando. Grazie a ciò rende più semplice imparare ad usare lo strumento di versionamento e dà una visualizzazione grafica dello storico della repository.

[qui ci va l'immagine dello screen di gitkraken](#)

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 \LaTeX

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