

Rapport d'activités générales des taches effectués.

Realisé par : Fatima zahra tijani

Table des matières

Introduction:	2
Scénario #1	3
1 : First Step : Immersion	3
2 : Second Step : La découverte	
3 : third step : Historique	5
Scénario2	6
1 : First Step : Tagging	6
2 : Second Step : Stashing and Saving work in Progress	6
Scénario 3	8
Second Step : Branching and merging sur GITHUB	8
Third step : compare pull Requests	10
CONCLUTION:	11

Introduction:

La gestion des workflows sous GIT/GITHUB sera la base de notre apprentissage en terme de la gestion de projet agile.

Un Scrum Master maintient la motivation de l'équipe au plus près des objectifs stratégiques. Il/Elle est le garant de l'amélioration continue des pratiques professionnelles. Il/Elle favorise une communication conviviale et un management visuel en gestion de projet agile.

En faisant preuve de transparence et d'inspections bienveillantes, il/elle est capable d'analyser rapidement le niveau de maturité des demandes de l'équipe et leur impact sur la gestion quotidienne du projet.

Scénario #1

1 : First Step : Immersion

Git : est un logiciel de gestion de versions décentralisé. C'est un logiciel libre créé par Linus Torvalds.

```
VourcodeDELEXTOP-0001901 WINAMA -/Desktop/project/demo

VourcodeDELEXTOP-0001901 WINAMA -/Desktop/project/git/
VourcodeDELEXTOP-0001901 WINAMA -/Desktop/project (master)

VourcodeDELEXTOP-0001901 WINAMA -/Desktop/project (master)

VourcodeDELEXTOP-0001901 WINAMA -/Desktop/project (master)

VourcodeDELEXTOP-0001901 WINAMA -/Desktop/project/demo

1 In VourcodeDELEXTOP-0001901 WINAMA -/Desktop/project/demo

2 In VourcodeDELEXTOP-0001901 WINAMA -/Desktop/project/demo

3 In VourcodeDELEXTOP-0001901 WINAMA -/Desktop/project/demo

3 In VourcodeDELEXTOP-0001901 WINAMA -/Desktop/project/demo

4 VourcodeDELEXTOP-0001901 WINAMA -/Desktop/project/demo (master)

5 In VourcodeDELEXTOP-0001901 WINAMA -/Desktop/project/demo (master)

5 Touch READEL and vourcodeDELEXTOP-0001901 WINAMA -/Deskt
```

git : (c'est un dossier caché situé à la racine du projet). Il contient l'historique des modifications des fichiers et la configuration de Git pour ce projet. git1.PNG

2 : Second Step : La découverte

```
## A STATE OF THE PRINCE OF THE PRINCE OF THE PRINCE OF THE OWN O
```

HEAD : est une référence au dernier commit de la branche de sortie actuelle. Vous pouvez considérer la tête comme la "branche actuelle".

LOGS : L'outil Git Log vous permet d'afficher des informations sur les validations précédentes effectuées dans un projet.

BRANCHES : Une branche dans Git est simplement un pointeur mobile léger sur l'un de ces commits. Le nom de branche par défaut dans Git est master

3: third step: Historique

Pour commencer cette étape, nous avons tout d'abord Affichez le dernier commit, toute en ajoutant l'option d'affichage de la hiérarchie de la branche, avec les commit et leur branche aussi. Nous avons fait ça avec la commande suivant :

```
Voucode@DESKTOP-62DG4RA MINGW64 -/Desktop/project/demo (master)
3 git show --oneline
6e7ad82 (HEAD -> master) le deuxeme commit
diff --git _A_1/tence.md b/Licence.md
new file mode 100644
index 00000000..e69de29

Voucode@DESKTOP-62DG4RA MINGW64 -/Desktop/project/demo (master)
5 git config --global alfas.historique show --oneline

Voucode@DESKTOP-62DG4RA MINGW64 -/Desktop/project/demo (master)
5 git config --list | grep alfas
alfas.historique=log
alfas.historique=show
```

```
Youcode@DESKTOP-62DG4RA MINGW64 -/Desktop/project/demo (master)

S git log
commit 667ads21004b6ba4321101753aad64c987b31d65 (HEAD -> master)
Author: achraf lachgar <achraflachgar456@gmail.com>
Date: Tue Nov 26 19:06:56 2019 +0100

le deuxeme commit

commit 932ac86f2092b604840dba07529ca226fc1900c8
Author: achraf lachgar <achraflachgar456@gmail.com>
Date: Tue Nov 26 19:03:40 2019 +0100

le premier commit

Youcode@DESKTOP-62DG4RA MINGW64 -/Desktop/project/demo (master)
5 git historitume 932ac86f2092b604840dba07529ca226fc1900c8
commit 932ac68f2092b604840dba07529ca226fc1900c8
commit 932ac68f2092b604840dba07529ca226fc1900c8
commit 932ac68f2092b604840dba07529ca226fc1900c8
commit 932ac68f2092b604840dba07529ca226fc1900c8
commit 932ac68f2092b604840dba07529ca226fc1900c8
commit of the commit of the committee of the committe
```

Scénario 2

1 : First Step : Tagging

```
youcode@DESKTOP-LKQJF63 MINGW64 ~/desktop/projects (master)
$ cd projects

youcode@DESKTOP-LKQJF63 MINGW64 ~/desktop/projects (master)
$ cd Projects/

youcode@DESKTOP-LKQJF63 MINGW64 ~/desktop/projects/Projects (master)
$ cd demo/

youcode@DESKTOP-LKQJF63 MINGW64 ~/desktop/projects/Projects/demo (master)
$ git tag -a V1.0 -m "REALEASE 1.0"

youcode@DESKTOP-LKQJF63 MINGW64 ~/desktop/projects/Projects/demo (master)
$ git tag
V1.0

youcode@DESKTOP-LKQJF63 MINGW64 ~/desktop/projects/Projects/demo (master)
$ S git tag
V1.0
```

2 : Second Step : Stashing and Saving work in Progress

```
oucode@DESKTOP-9GDI9BI MINGW64 ~/Desktop/projects/demo (master)
  vim README.md
 oucode@DESKTOP-9GDI9BI MINGW64 ~/Desktop/projects/demo (master)
  git stash
varnings: LF will be replaced by CRLF in .gitignore.
The file will have its original line endings in your working directory
Saved working directory and index state WIP on master: ff70174 branche bien faite corigier
 oucode@DESKTOP-9GDI9BI MINGW64 ~/Desktop/projects/demo (master)
 git stash list
tash@{0}: WIP on master: ff70174 branche bien faite corigier
 oucode@DESKTOP-9GDI9BI MINGW64 ~/Desktop/projects/demo (master)
 git status
n branch master)
Untracked files:
  (use "git add <file>..." to include in what will be committed)
nothing added to commit but untracked files present (use "git add" to track)
 oucode@DESKTOP-9GDI9BI MINGW64 ~/Desktop/projects/demo (master)
 vim Licence.txt
/oucode@DESKTOP-9GDI9BI MINGW64 ~/Desktop/projects/demo (master)
5 git commit -a -m'first commit'
varning: LF will be replaced by CRLF in Licence.txt.
The file will have its original line endings in your working directory
[master 029ca36] first commit
1 file changed, 1 insertion(+)
 oucode@DESKTOP-9GDI9BI MINGW64 ~/Desktop/projects/demo (master)
 git stash pop
n branch master
Changes not staged for commit:

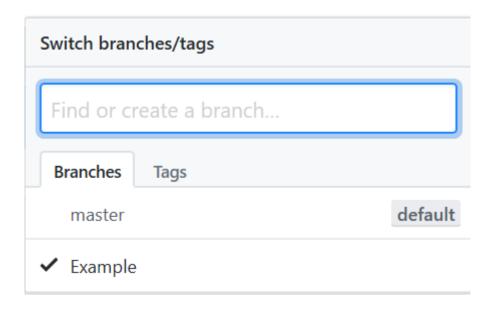
(use "git add <file>..." to update what will be committed)

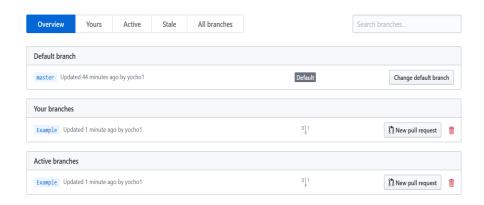
(use "git checkout -- <file>..." to discard changes in working directory)
no changes added to commit (use "git add" and/or "git commit -a")
Dropped refs/stash@{0} (b3e6ffef3456260c29466680366c6047aad5bdc3)
 oucode@DESKTOP-9GDI9BI MINGW64 ~/Desktop/projects/demo (master)
 git stash pop
o stash entries found.
```

git stash: temporairement les modifications que vous avez apportées à votre copie de travail afin que vous puissiez travailler sur autre chose, puis revenir et les réappliquer plus tard

Scénario 3

Second Step: Branching and merging sur GITHUB





```
YoucodeDESKTOP-62DG4RA MINGW64 -/Desktop/Projects/Monsiteweb-local (master)

5 git branch annulation

YoucodeDESKTOP-62DG4RA MINGW64 -/Desktop/Projects/Monsiteweb-local (master)

5 cd initializr/

YoucodeDESKTOP-62DG4RA MINGW64 -/Desktop/Projects/Monsiteweb-local/initializr (master)

5 vim index.html

YoucodeDESKTOP-62DG4RA MINGW64 -/Desktop/Projects/Monsiteweb-local/initializr (master)

5 git checkout annulation

W initializr/index.html

YoucodeDESKTOP-62DG4RA MINGW64 -/Desktop/Projects/Monsiteweb-local/initializr (annulation)

5 vim i

YoucodeDESKTOP-62DG4RA MINGW64 -/Desktop/Projects/Monsiteweb-local/initializr (annulation)

5 vim index.html

YoucodeDESKTOP-62DG4RA MINGW64 -/Desktop/Projects/Monsiteweb-local/initializr (annulation)

5 vim index.html

YoucodeDESKTOP-62DG4RA MINGW64 -/Desktop/Projects/Monsiteweb-local/initializr (annulation)

5 git commit -a - m 'annulation de Hi'

1 file changed, 1 deletion(-)

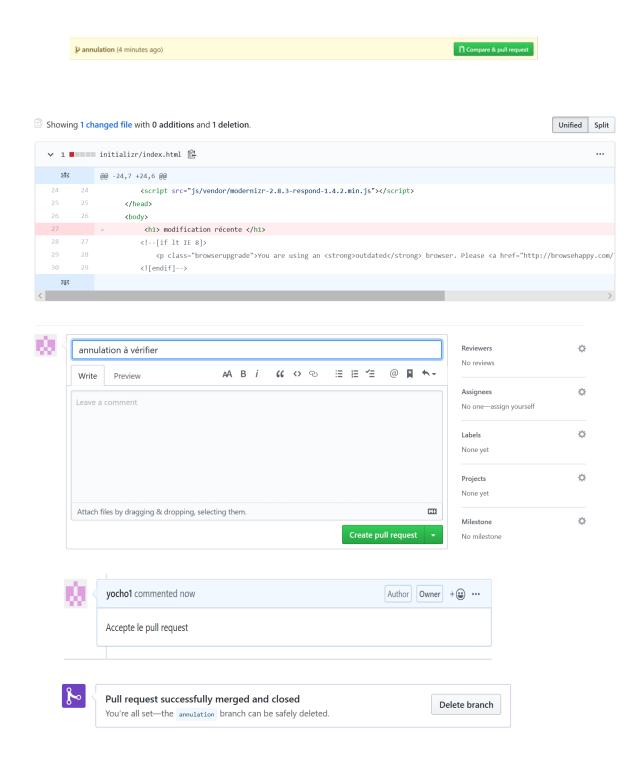
YOUCOdeDESKTOP-62DG4RA MINGW64 -/Desktop/Projects/Monsiteweb-local/initializr (annulation)
```

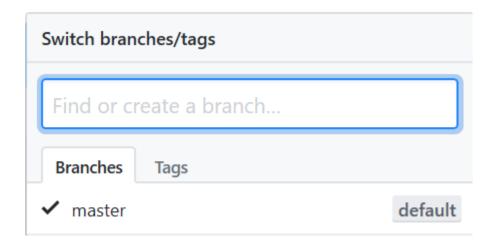


```
VoucadeDESKTOP-6204RA MINGM64 -/Desktop/Projects/Monsiteweb-local/initializr (annulation)
3 git show
commit ed210b5e0407d34d040774227df6888b9b5430 (MEAD -> annulation)
Author: achraf lachpar-4schraflachpar-4568gmail.com>
Date: Thu Nov 28 00140100 2019 +0100
annulation de H1

diff --git a/initializr/index.html b/initializr/index.html
index dbaad97..8db9e35 100644
--- a/initializr/index.html
H= b/initializr/index.html
H= b/initializr/index.html
H= b/initializr/index.html
H= b/initializr/index.html
H= c/r/initializr/index.html
H= c/r/i
```

Third step: compare pull Requests





CONCLUTION:

J'ai appris beaucoup de plaisir à en apprendre plus sur Git et à débloquer ses secrets. Il est très performant du fait d'avoir été écrit en C et optimisé pour le noyau Linux. La notion de Git Branching est terriblement efficace et permet de construire un projet de bout en bout de façon très simple et intuitive en minimisant les crises de conflits. Avec Git vous pouvez faire presque tout ce que vous voulez. La gestion des tags, branches, et merge est à la limite de la perfection en des temps records .

De plus, Github contribue largement à la notoriété de Git en fournissant un service d'hebergement de code open source permettant la visualisation du code, en ajoutant le côté social et relationnel.