

# Outils Libres pour le développement logiciel

## 1. Définir le workflow gitflow, pourquoi on l'utilise

Le workflow gitflow permet à une équipe de travailler sur un projet commun en développant des fonctionnalités sur différentes branches .

## 2. Quels sont les avantages du workflow gitflow

Les avantages sont lorsqu'il faut effectuer un travail en groupe, faire des essais, des fonctionnalités à part, et le versioning, c'est-à-dire de retrouver une version antérieure après avoir fait un pull request.

## 3. Quels sont les inconvénients du workflow gitflow

Les inconvénients du workflow gitflow sont les petits et longs projets, c'est-à-dire, qu'ils ne s'adaptent très bien pour les petits projets vu le nombre de branches. Tandis que sur les longs projets, l'inconvénient est s'il y a un responsable de projet qui doit valider les modifications avant de les insérer.

## 4. Définir et donner l'utilité des branches : Feature, Hotfix, Release, Develop, Master

Feature : La branche feature sert à développer les différentes fonctionnalités. Une branche feature correspond à une fonctionnalité. Lorsque la fonctionnalité a fini d'être développé sur la branche feature, elle merge dans la branche develop.

Hotfix : La branche hotfix a le même principe que la branche release cependant elle est basée sur la branche master.

Release : La branche release est établie sur la branche develop. Elle sert à corriger les bugs.

Develop : La branche develop aura tous les commits effectués, comme un historique.

Master : La branche master sert à avoir le projet final où les différentes fonctionnalités auront été corrigées avant. Elle contiendra les commits, une version synthétisée de la branche develop.

## 5. Donner les commandes git pour créer un tag, sachant que vous êtes sur la branche Develop

- git tag -a <tag\_name> -m "msg "
- git push -tags

## 6. Vous êtes sur une branche Feature en train de finaliser un développement, on vous informe qu'il y a un bug de prod à corriger très rapidement. Donner les commandes git pour corriger le problème de prod en respectant le workflow git

- git checkout -b hotfix-x.x.x master
- git commit -m "msg"
- git checkout master

- git merge --no-ff hotfix-x.x.x
- git tag -a x.x.x -am "message"
- git checkout develop
- git merge --no-ff hotfix-x.x.x -à la fin on supprime la branche du hotfix
- git branch -d hotfix-x.x.x

**7. Donner les commandes git à exécuter après la validation de la branche release pour passer en prod**

- git checkout master
- git merge release <version>

**8. À quoi sert la commande git stash, donner la commande qui permet d'avoir un retour arrière de git stash**

La commande git stash permet d'enregistrer provisoirement le travail actuel, et de travailler sur une autre tâche.

- git stash pop

Amélie PSYCHE