

Nom : Lebdiri Merchiche

Prénom : Melissa

Travail_à_rendre_git

- 1) **gitflow** : c'est une architecture de branches, elle est utilisée pour faire la gestion des branches ça nous permet de faire des développements et de gérer le lien entre la branche master de production et la branche master des développements.

- 2) **Les avantages du workflow gitflow :**

Cette commande elle dispose d'une grande simplicité, elle permet de travailler en équipe, ça permet au développeur de mettre leur nouvelle fonctionnalités sur la branche de développement une fois que la période de développement ou le sprint est terminé ce mécanisme, nous permet de mettre directement les nouvelles fonctionnalités développer sur le projet qui est en production, notamment elle permet de créer plusieurs fonctionnalités en parallèle, ça facilite de faire une gestion des tâches et développer des nouvelles fonctionnalités.

- 3) **Les inconvénients du workflow gitflow :**

Ce processus demande beaucoup du temps et ne s'applique pas sur les projets en effet, pour définir la bonne stratégie de développement il faudra faire une longue analyse du projet et dans la plupart des cas et surtout des projets où les fonctionnalités ne sont pas finies à l'avance cela pourrait poser un problème d'analyse, dispose d'un historique très difficile à suivre, nous pouvons juste l'utiliser dans un modèle de version traditionnel, difficile à gérer pour le modèle déploiement continu version traditionnel.

- 4) **Définition et utilisation des branches :**

Master : il s'agit de la version finale qui est en production.

Feature : c'est les nouvelles branches que les développeurs créent en fonction de leur développement ça peut être considéré comme un brouillon.

Release : c'est la branche dans laquelle on crée des nouvelles versions ou des tags pour le projet.

Hotfix : c'est la branche où les problèmes et les bugs sont corrigés.

Develop : c'est la branche qui contient des nouvelles fonctionnalités développées avant de les transmettre au master.

- 5) **Les commandes git pour créer un tag :**

git flow release start numéro de tag

- 6) **Les commandes git pour corriger le problème prod :**

git flow hotfix start nom de la branche hotfix

Et là on corrige le bug une fois que ces dernières sont corrigées et qu'on est prêt à remettre le code en production il faudrait lancer la commande suivante :

git flow hotfix finish nom de la branch hotfix

7) Les commandes de git à exécuter après la validation de la branche release pour passer en prod :

git flow release finish numéro de tag

8) A quoi sert la commande git stash :

Git stash : il sauvegarde les modifications non approuvé dans un stash.

Donner la commande qui permet d'avoir un retour en arrière de git stash

git stash pop