

GIT FLOW

Une Version Alternative de Git

- adaptation spécifique

- modèle de branchage

(Vincent Driessen)

<http://nvie.com/posts/a-successful-git-branching-model/>

- nouvelles commandes

Un peu d'Histoire....

- Stratégie novatrice pour la gestion des branches git.
- Désuétude (tronc).

Adapté?

- Cycle de livraisons planifiées.
- Livraison continue(petites itérations).
- Installation sous Windows <https://git-scm.com/download/win>.
- Installation sous OSX: » brew install git-flow«.
- Exécution: « git flow init ».

Modèle de branchage

- Branche « Main »(H.officiel).
- Branche « Develop »(fonctionnalités).
 - ✓ « git branch develop »
« git push -u origin develop ».
- Branche develop automatique(sur dépôt existant).
 - « Git flow init »
- Branche Feature(chaque fct).
 - ✓ Sans git flow: »git checkout develop »
« git checkout -b feature_branch »
 - ✓ Avec git flow: »git flow feature start feature_branch »
- Branche livraison(+/- fonctionnalités).
- Branche Hotfix(patch rapide).

Pour résumer....

1. Une branche develop est créée à partir de main.
2. Une branche release est créée à partir de la branche develop
3. Des branches feature sont créées à partir de la branche develop
4. Lorsqu'une fonctionnalité est terminée, elle est mergée dans la branche develop.
5. Lorsque la branche release est terminée, elle est mergée dans la develop et dans main.
6. Si un problème est détecté dans la branche main, une branche hotfix est créée à partir de main.
7. Une fois la branche hotfix terminée, elle est mergée dans develop et dans main.

Différents Liens

- <https://www.atlassian.com/fr/git/tutorials/comparing-workflows/gitflow-workflow>
- [https://danielkummer.github.io/git-flow-cheatsheet/index.fr FR.html](https://danielkummer.github.io/git-flow-cheatsheet/index.fr_FR.html)