Table des matières

[1 Qu’est-ce que Git ? 2](#_Toc93408417)

[1.1 Contrôle de version Git 3](#_Toc93408418)

[1.1.1 Git est un outil de qualité 3](#_Toc93408419)

[1.1.2 Git est un projet open source de qualité 3](#_Toc93408420)

[2 GitHub 3](#_Toc93408421)

[2.1 Fonctionnalités 3](#_Toc93408422)

[3 L’intérêt pour un projet informatique 3](#_Toc93408423)

[3.1 L’intérêt pour notre projet informatique 3](#_Toc93408424)

[4 Glossaire 3](#_Toc93408425)

# Qu’est-ce que Git ?

Git est de loin le système de contrôle de version le plus largement utilisé aujourd'hui. Git est un logiciel de gestion de version.

À l'origine, il a été développé en 2005 par Linus Torvalds, le créateur bien connu du noyau du système d'exploitation Linux.

De plus en plus de projets logiciels reposent sur Git pour le contrôle de version, y compris des projets commerciaux et en open source.

Les développeurs qui travaillent avec Git sont bien représentés dans le pool de talents disponible, et la solution fonctionne bien sur une vaste gamme de systèmes d'exploitation et d'environnements de développement intégrés (IDE).

Par sa structure décentralisée, Git illustre parfaitement ce qu'est un système de contrôle de version décentralisé (DVCS).

Plutôt que de consacrer un seul emplacement pour l'historique complet des versions du logiciel comme c'était souvent le cas dans les systèmes de contrôle de version ayant fait leur temps, comme CVS et Subversion (également connu sous le nom de SVN), dans Git, chaque copie de travail du code est également un dépôt qui contient l'historique complet de tous les changements.

En plus d'être décentralisé, Git a été conçu pour répondre à trois objectifs : performances, sécurité et flexibilité.

## Contrôle de version Git

Git est la solution idéale pour la plupart des équipes de développement actuelles. Si chaque équipe est différente et doit faire sa propre analyse, voici les principales raisons pour lesquelles le contrôle de version avec Git est l'option à privilégier

### Git est un outil de qualité

Git offre les fonctionnalités, les performances, le niveau de sécurité et la flexibilité dont la plupart des équipes et des développeurs ont besoin. Ces avantages de Git sont repris en détail ci-dessus. Lorsqu'elles les comparent à d'autres solutions, beaucoup d'équipes préfèrent Git.

### Git est un projet open source de qualité

Git est un projet open source bien pris en charge, qui a bénéficié d'une gestion solide pendant plus de dix ans. Les mainteneurs du projet ont su faire preuve de jugement et ont adopté une stratégie réfléchie pour répondre aux besoins à long terme de leurs utilisateurs, en publiant régulièrement de nouvelles versions pour améliorer l'ergonomie et les fonctionnalités. La qualité du logiciel open source est évidente, et un nombre incalculable d'entreprises misent sur cette qualité. Git est largement approuvé et compte une vaste base d'utilisateurs. La documentation est excellente et abondante : elle inclut des livres, des tutoriels et des sites Web dédiés. Mais ce n'est pas tout : vous trouverez également des podcasts et des tutoriels vidéo. La mise en open source réduit les coûts pour les développeurs amateurs, puisqu'ils peuvent utiliser Git gratuitement. Pour les projets open source, Git est incontestablement le successeur des anciennes générations de systèmes de contrôle de version ayant fait leurs preuves, comme SVN et C

# GitHub

GitHub est une service web d’hébergement et de gestion de développement de logiciels, utilisant le logiciel de gestion de version Git.

Le nom GitHub est composé du mot « git » faisant référence à un système de contrôle de version open-source et le mot « hub » faisant référence au réseau social bâti autour du système Git, mais aussi à une plate-forme de correspondance qui est appelée en anglais un « hub ».

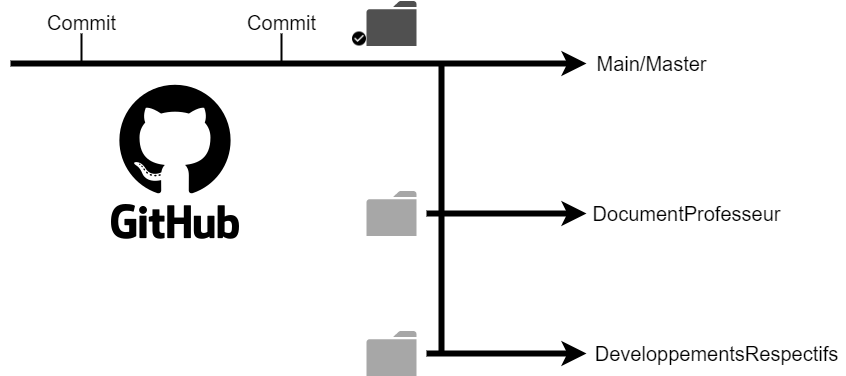
## Fonctionnalités

GitHub est centré vers l'aspect social du développement. En plus d'offrir l'hébergement de projets avec Git, le site offre de nombreuses fonctionnalités habituellement retrouvées sur les réseaux sociaux comme les flux, la possibilité de suivre des personnes ou des projets ainsi que des graphes de réseaux pour les dépôts (en anglais « repository »). GitHub offre aussi la possibilité de créer un wiki et une page web pour chaque dépôt. Le site offre aussi un logiciel de suivi de problèmes. GitHub propose aussi l'intégration d'un grand nombre de services externes, tels que l'intégration continue, la gestion de versions, badges, chats basés sur les projets, etc.

# L’intérêt pour un projet informatique

Il est intéressant pour un projet informatique d’utiliser Git et GitHub. Car avec le « versionning » vous pourrez identifier la modification responsable de votre bug, qui l'a provoqué et quand. Il est également possible de restaurer la partie concernée pour rétablir le fonctionnement de votre application. Travailler chacun de votre côté et rassembler tout votre travail en commun facilement.

## Schémas de notre projet informatique



# Glossaire

Commande les plus utilisées durant notre projet :

**$git add NomDuFichier**

(Pour ajouter un fichier dans le dépôt local)

Ou **$git add \***

(Si on veut ajouter tous les fichiers disponibles dans le dépôt local)

**$git commit -m 'NomExplicatif’**

(Pour prendre en photo tous les documents qui ont été ajoutés dans le dépôt local, ce qui permet de sauvegarder les versions qui nous intéresse. Y ajouter un nom qui nous indique pourquoi le commit a été réalisé.)

**$git log --oneline**

(Permet de vois les différents commits réalisés et ainsi voir les différentes versions)

**$git reset**

(Qui permet de revenir à un commit réaliser (soit une version en arrière))

**$git branch**

(Permet d’afficher les branchs (ou les différents chemins) qui ont été créés)

$git switch NomDeLa