Utiliser les commandes Github simplement
Pour plus d'infos : https://git-scm.com/book/fr/v2

Quelques petits conseils sur des commandes simples à utiliser dans github pour éviter les problèmes ou les fichiers qui ne s'envoient pas :

1. Aller dans le bon répertoire

Si vous n'avez pas de dossier « Twizy » sur votre PC, il faut utiliser la commande : git clone git@github.com:NadraniYassine/Twizy.git

C'est une connexion grâce au lien SSH qui demande donc d'avoir lié sa clé ssh à son compte github (voir le premier paragraphe du TD github ACL).

Une fois ceci fait, le dossier /Twizy devrait être crée. Toutes les autres commandes demandent d'être dans le dossier, on s'y rend grâce à la commande cd Twizy .

Si la commande ne fonctionne pas : soit le dossier n'existe pas et il faut le recloner, soit vous ne vous trouvez pas dans le bon répertoire, il faut dans ce cas-là spécifier le chemin complet dans la commande cd (tous les dossiers depuis le répertoire de base qui est ~) cd ~/...../Twizy [ex cd ~/Documents/Twizy]

Une fois dans ce dossier, la commande git pull permet de récupérer la version la plus à jour du dossier telle qu'elle se trouve sur Arche, il faut impérativement pull avant tout car si le dossier sur lequel vous travaillez n'est pas le même que celui sur git, vous risquez d'avoir une erreur ensuite.

2 . Ajout de documents

Vous pouvez ensuite ajouter les documents sur lesquels vous avez travaillé dans le dossier à la bonne place, une fois ceci fait il faut faire les commandes :

git add [NOM DU DOCUMENT/DOSSIER]

git commit -m « message qui explique ce que vous avez fait »

git push

[ex : git add Extras / git commit -m « Ajout document tuto github / git push]

Si vous oubliez le -m après le git commit, vous allez être dirigé vers une fenêtre vim où vous allez devoir rentrer le message explicatif. Pour naviguer dans vim, appuyez sur i permet de rentrer en mode édition où vous pouvez rentrer votre message. Une fois ceci fait, vous pouvez échap pour retourner en mode commande et :x pour enregistrer et quitter.

3. Le Tag

Attention, le tag ne se fait que lorsque la version mise en place sur git est fonctionnelle, il faut donc que ce soit validé par tout le monde et la phase de tests.

Ne pas tag à chaque fois que vous mettez quelque chose sur git.
En règle générale, ne pas tag.

Pour faire un tag, on utilise la commande git tag -a [version] -m « [message] »
[ex : git tag -a v1.1 -m « version fonctionnelle ligne de cmd »]

Encore une fois, ne pas mettre le -m ouvre l'éditeur vim, se référer au point 2. La version est de la forme X.Y avec X = numéro du sprint et Y = chiffre suivant la dernière version active.

Pour avoir l'historique des versions on utilise git tag. Une fois ceci fait, il reste à push avec la commande particulière : git push origin [version]
[ex : git push origin v1.1]

4. Bug

Si en faisant un push vous avez un message d'erreur qui vous dit que vous ne travaillez pas sur la bonne version du dossier, c'est que quelqu'un a mis à jour sur github un document que vous essayez de push, il faut peut-être add seulement le document sur lequel vous venez de travailler si ce n'est pas celui qui a été modifié. Sinon, c'est un problème. Dans ce cas-là vous ne pouvez plus pull.

La manière la plus simple de fonctionner est de sortir du document cd (pas besoin d'argument supplémentaire, cela ramène au répertoire home) et de supprimer le document « Twizy » pour revenir à l'étape 1. Ne pas oublier évidemment de garder une copie du document sur lequel vous avez travaillé avant de tout supprimer.