**GIT – Generic Workflow**

Overview of GIT command:

\*\*\* note: master or main is pretty similar. Most company will use master as their main branch

1. git fetch upstream

2. git merge upstream/master

3. git push --set-upstream origin master

4. git checkout -b new\_feature

5. git add files

6. git commit

7. git push --set-upstream origin new\_feature

8. git checkout master

9. git merge --no-ff --verify-signatures new\_feature

10. git push --set-upstream origin master

11. git branch -D new\_feature

12. git push --set-upstream origin :new\_feature

1. git fetch upstream

Before you begin, update your remote-tracking branches (this operation never changes any of

your local branches).

2. git merge upstream/master

Merge your updated remote-tracking branches into your master branch.

3. git push --set-upstream origin master

Update your master’s origin on Github.

4. git checkout -b new\_feature

Create a new branch, new\_feature branch, and switch to it.

5. git add files

After some additions, modifications, or deletions are made, stage the files.

6. git commit

Sign your committed change.

7. git push --set-upstream origin new\_feature

Push your new\_feature changes into Github.

7. git checkout master

Switch to your master branch.

8. git merge --no-ff --verify-signatures new\_feature

Merge your new\_feature branch into your master branch with the recursive strategy and

sign the merge.

9. git push origin master

Push your master branch to its origin on Github.

IMPORTANT If your code works on production flawlessly, clean up with steps 10 & 11.

10. git branch -D new\_feature

If your new\_feature branch works flawlessly. Delete your local new\_feature branch.

11. git push --set-upstream origin :new\_feature

Delete your new\_feature branch from its origin on Github.

**GIT – Push to Development**

# Update your remote-tracking branches

cd $DOCUMENT\_ROOT

git checkout master

git fetch --prune upstream

git merge upstream/master

git push --set-upstream origin master

Step 1:

git checkout -b feature\_branch

git add files

git commit

git push --set-upstream origin feature\_branch

Step 2 (Optional):

If you need to amend/add onto your previous commit

git add files

git commit --amend

git push --set-upstream origin feature\_branch -f

\*\*\* NOTE:

*If you want to remove the file from the Git repository and the filesystem, use:*

git rm file1.txt

git commit -m "remove file1.txt"

*But if you want to remove the file only from the Git repository and not remove it from the filesystem, use:*

git rm --cached file1.txt

git commit -m "remove file1.txt"

And to push changes to remote repo

git push origin branch\_name