As I researched how Git could be utilized in a team environment, I found some very good articles that concisely explained the most important features of Git and how each feature fit together in the context of collaboration.  We start with git clone because this will allow us to create a local copy of our project from the remote repository.  Once we have this, we can use the branch command to create a new branch to work on so that we can make changes to our project without interfering with other developers on the project.  After creating the branch, we can use checkout to switch back and forth between branches.  While working on a branch, we can use the add command to stage our changes and then commit the changes and take a snapshot of our work.  Most of these commands also allow us to add extra parameters depending on whether we need to use additional features.  A very simple example of this is the use of "-m" with git commit to allow us to add a commit message directly to commit without opening a word processing document, but this is just one example.  After committing, we can push the changes from our branch out to the remote repository without interfering with the master branch.

The sources I found also emphasized the importance of pulling from the remote repository on a regular basis so that any conflicts that need to be resolved will be simple ones.  Of course there are many other commands that we have learned that I have not yet mentioned like log and status but if we are discussing git in the context of teamwork, the commands I mentioned initially are, in my opinion, the most useful for promoting Agile methodologies.  With respect to Continuous integration, merge and pull come to mind as extremely relevant.  I also found an article that went in-depth into the usefulness of branches and provided a very good visual representation of how various branches could support Agile methodologies by using different branches for fixes, releases, development branches, and feature branches.  A team following this model of branch use could quickly push changes out to the end user of their product.  I have attached a link to the sites I found most helpful.

[https://www.fosslinux.com/133673/how-to-use-git-commands-to-collaborate-with-others.htmLinks to an external site.](https://www.fosslinux.com/133673/how-to-use-git-commands-to-collaborate-with-others.htm)

[https://medium.com/novai-devops-101/top-4-branching-strategies-and-their-comparison-a-guide-with-recommendations-21071e1c472aLinks to an external site.](https://medium.com/novai-devops-101/top-4-branching-strategies-and-their-comparison-a-guide-with-recommendations-21071e1c472a)