Github Workflow

Non-Training Program Workflow:

1. Project Manager or Team Lead creates a team on mdb github account with all team members included.
2. Project Manager or Team Lead creates project repo under the mdb github org, and assigns their team to the repo.
3. Project Manager or Team Lead assigns stories to members
    a. Make each story as specific as possible (ex: implement this function).
    b. In the description for the story, specify acceptance criteria. This should include all the details a developer would need to complete the story.
    c. Imagine the developer knows the bare minimum about your project so be as descriptive as possible.
    d. Assign each story to a person.
    e. These stories should take multiple hours to plan if done well.
    f. Essentially, you are coming up with a very detailed spec for your codebase before you start coding!
    g. We recommend using Kanban view & task list in Freedcamp
    h. You can split the tasks into different task groups (ex: frontend, backend, design, etc). Make sure to give each story in your Freedcamp a story number (ex: Frontend-23) and a title.
4. Members create story branches for each task they are assigned.
    a. Never merge story branches.
    b. Name of each branch should be brief name of story (all lowercase with dashes if need be).
5. Members create pull requests when their story branch has been bug tested and is fully complete.
6. Project Manager or Team Lead code reviews pull request and merges with master. They may do additional bug testing if they like.

Training Program Workflow:

1. Instructors create repos under the mdb github org, and invites all respective training program members to the repos.
    a. Each repo is for each mini-project/assignments
2. Instructors may put project/assignment skeleton on master branch or just keep repo blank (sometimes prevents merge conflicts)

3. Training Program members push to a branch named accordingly to their group. For example: krishnan-aneesh would be a branch under sp17AndroidProj1, and aneesh and I would individually commit our individual work to that branch.
4. When members are ready to submit they submit pull request from their groups branch to master
5. Instructors and Senior Devs can code review pull requests (without merging) and leave feedback in comments.
6. VP of New Members is responsible for putting scores in MDB Central

Example Workflow

Let's say you're beginning a story that you are assigned to. First, pull from master using:

```
git pull origin master
```

Then, create a branch for your story and checkout:

```
git branch Frontend-23
git checkout Frontend-23
```

When you make progress on your story, push to your story's branch. In Freedcamp, move the story into the "In Progress" stage.

```
git add .
git commit -am "implemented helloWorld()"
git push origin Frontend-23
```

When you finish your story, go to github and make a pull request for your story. Request your project manager to do a code review. Even if there is no code to fix, project manager should comment on your pull request to make sure it was reviewed. They can say something as simple as "good work, no fixes needed".

If there are changes requested in the code review, make the changes, push to your branch, and make a new pull request.

Merge the pull request with master and ensure the project works with the merged code. Tell your teammates to pull from master so they have the most updated version of the project. In Freedcamp, you can now move this story to done.

Helpful Links

http://rypress.com/tutorials/git/tips-and-tricks

https://github.com/git-tips/tips
https://git-scm.com/book/en/v1/Git-Basics-Tips-and-Tricks
http://freedcamp.com