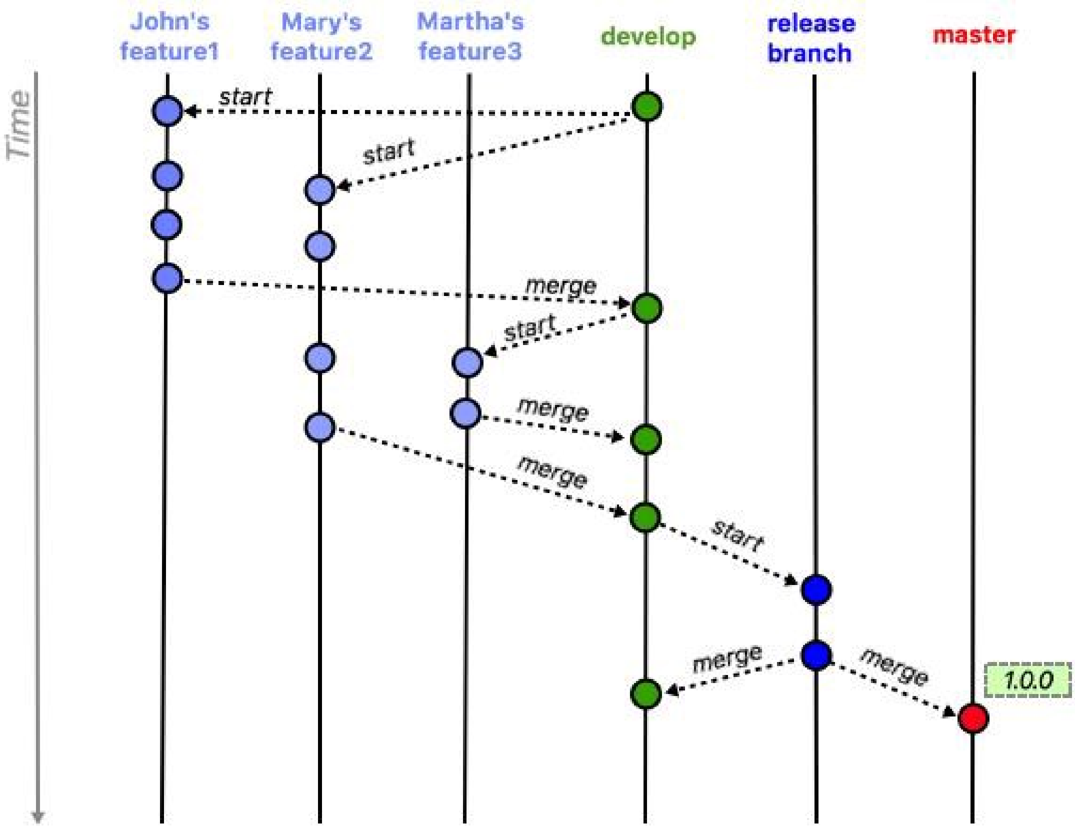
Feature branches

Remember this diagram?



https://medium.com/empathyco/git-flow-applied-to-a-real-project-c08037e28f88

So far, we have tackled and created the staging and development branches. In this section, we will discuss feature branches, the ones our developers use. Here: John, Mary, and Martha.

Feature branches

A feature branch is created by developers for every piece of work they decide to contribute. It can be created to contribute a big bulk of work, a bug fix, modify or enrich documentation…you name it! This is the starting point in the development process.

When a developer decides to work on an issue or a feature request, he/she starts by creating a feature branch. Depending on the branching strategy, it can “start from” the **development** or the **main** branch. In our case, we will start from the **development** branch.

In order to position yourself on a branch (before creating a new one for example), use the following command:

```
git checkout branch_name
```

Suppose we were a contributor to our project and we decided to work on updating file1.txt. We need to add a new line to the file: “New line representing the new feature.”. Again, we still do not know how to code so we are just modifying text files to illustrate how Git works. In a real development project, we would be modifying code. Those modifications might have serious implications!

▼ Position yourself on the **development** branch and create a feature branch called **file1-update**:

```
git checkout development
git checkout -b file1-update
```

▼ Make the necessary modifications to file1.txt, check that the file has been modified, check the modifications then commit them to the newly created branch (using the message: “feat: first new feature”)

```
git status
git diff file1.txt
git add file1.txt
git commit -m "feat: first new feature"
```

▼ Then push your work to the remote repository. This way, you will have a backup in case something happens to your local computer. Also, when your work is done, you will be able to show it to others and request it to be merged into the project (a simple git push will suffice for your next pushes of this branch):

```
git push --set-upstream origin file1-update
```

Note: there are several feature branch naming conventions you can choose from. Check out this link for example. Here, we are keeping things simple.