Natalie Belmonte
14 November 2023
Data Structures and Algorithms: Using a Git Workflow

GitHub repositories are a useful tool for program developers that allow them to easily collaborate on a single project. Repositories are very adaptable, and as such can be used in various ways. In order to maintain consistency, a developer team will establish a workflow at the beginning of a project. A workflow is a "...recommendation for how to use Git to accomplish work in a consistent and productive manner" ("Git…" 1). Depending on factors such as team size, project complexity, and release schedule, workflows can be as unique as the projects that they are applied to. Though developers can implement any guidelines they feel they need for a project, there are public Git workflows that are available for use.

Primarily, a workflow allows a team of developers to smoothly implement changes to their programs. Additionally, workflows should not make mistakes more difficult to undo or be more restrictive than what is needed. The most popular Git workflow, which uses a centralized repository, has each developer clone their own local version of the project. This allows them to make changes independently of the rest of the project before pushing them to the central repository.

"Pushing" changes requires the changes to be committed to the local repository. Committing code acts as confirmation that the code segment is completed, without any major errors. Then, the code can be pushed to the central repository, sharing it with all other developers and merging it into the preexisting program. This process allows multiple developers to edit the same sections of code, without worrying about their changes contradicting each other.

If a commit conflicts with what is already written in the central repository, the code will not be pushed. This is known as a merge conflict. Often, this occurs when a developer writes code, then tries to publish it shortly after the central repository is updated. To resolve this conflict, the code would need to be adapted with the updated project in mind.

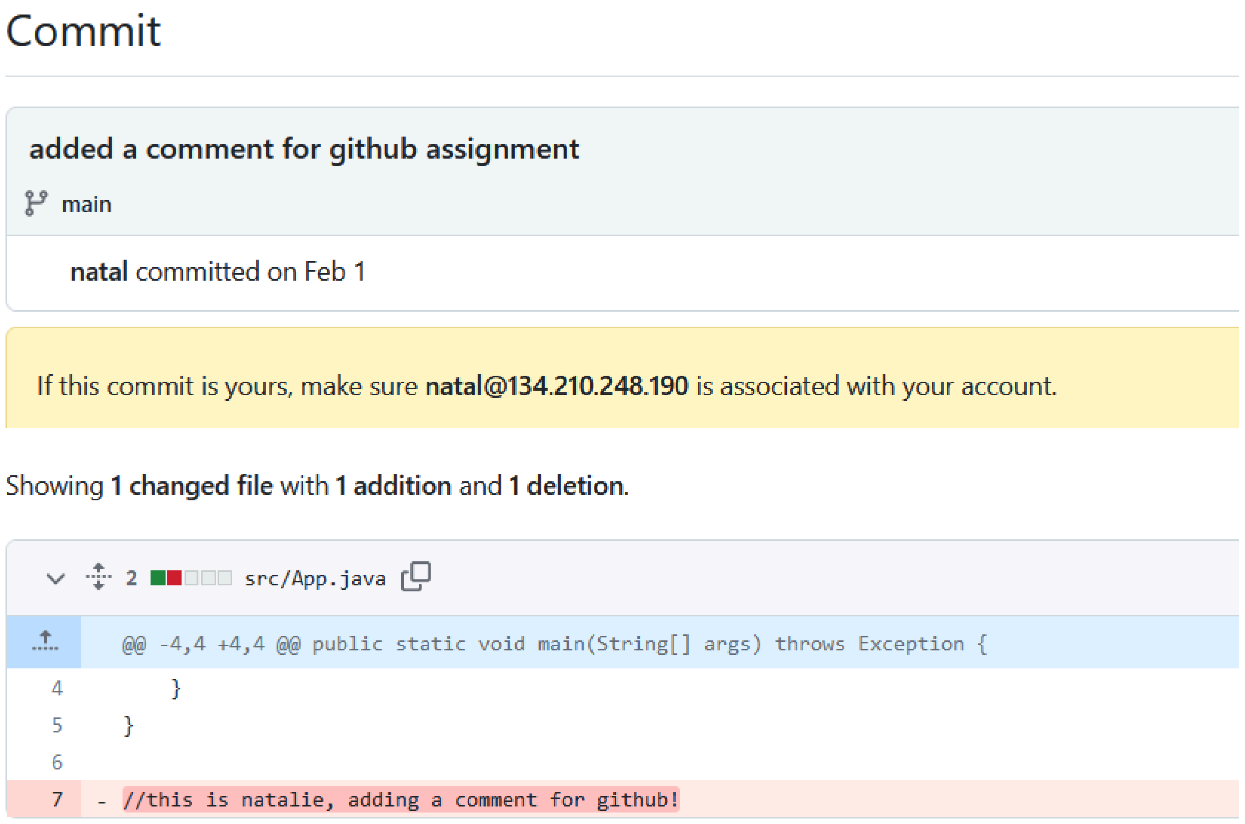
The prioritization of the central repository is used to keep the history of the program linear, making it easier to identify where errors may have come from. However, because the number of merge conflicts will increase with the number of developers, the Centralized Workflow is best for relatively small developer teams. Many other Git workflows are modified versions of the Centralized Workflow. Some, such as the Feature Branch Workflow, which divides each feature of a program into different sections, still utilize a central repository.

That being said, there are plenty of workflows that place less emphasis on a central repository, or do not use them at all. Gitflow, for example, is a feature branch-style workflow that determines what each branch is used for and how it should interact with any others. Once completed, each branch can be merged into the central repository. A Forking Workflow, on the other hand, allows for each developer to use both their local repository and their own public one on the project's server. Branching can be extremely useful for organizing complex programs. Generally, it is beneficial to keep branches separate from the main project for as little time as possible. As time goes on, the possibility for merge conflicts increases.

Using temporary branches is important for any Git workflow. Similarly, an effective workflow will minimize the need to revert the program, but still allow for reverts to be executed without much issue. There is no singular perfect Git workflow, as each team and project will have different needs. However, the standardization of the development process that comes from using a workflow makes them essential tools for any large-scale projects.

Works Cited

"Git Workflow: Atlassian Git Tutorial." *Atlassian*, 2023, www.atlassian.com/git/tutorials/comparing-workflows.



Commit

added a comment for github assignment

main

natal committed on Feb 1

If this commit is yours, make sure **natal@134.210.248.190** is associated with your account.

Showing **1 changed file** with **1 addition** and **1 deletion**.

```
2 src/App.java
@@ -4,4 +4,4 @@ public static void main(String[] args) throws Exception {
        }
    }

-   //this is natalie, adding a comment for github!
```

Commit message from a classmate's repository on 1 February 2024.