GitHub is one of the most used tools by software developers around the world. What makes GitHub such a beneficial tool for most development teams is the collaboration that it provides. This is due to teams being able to create workflows to work on projects. These workflows allow for code to be easily created, reviewed, and merged by a team.

For a workflow to be created a team must first make a repository to store the many components of the project in. This repository can then be shared among all the members of a team, which allows them to make changes to the project that all members of a team can see. It is then that each team member would create a branch to do their own work without interfering with everybody else’s work.

Now that everybody has a branch, they are allowed to work on their part of the project wherever they are, no more worrying about people not being in the office so their part of a project will not be done on time. This is all thanks to the Git system which allows users to track all the changes happening to their code, as well as copy it to their own computer. For this reason, many software companies are allowed to be fully remote.

So, all the members of a team are hard at work writing code but how does it all come together to make a fully function piece of software. The answer is committing. Committing is the way that code get added to a user’s branch. With each commit a message must be provided, this allows users to keep a log of all changes that took place between the last commit and the new one. Committing only applies to a single person’s branch, to get the code in the main branch of the repository a push statement is used. This sends all other team members a pull request to let them know that someone is trying to make changes to the main branch. It is now that all team members have the chance to manually look over the work that has been done to ensure that it is up to the standards that the team has in place.

After the code has been successfully pushed to the main branch in the repository, similarly to when changes are committed to a branch, there is a timestamp of when the code was pushed and a message detailing the changes made for everyone to see. Now the code in the main branch is updated so that when anymore changes are made to the main branch GitHub will automatically check to see if the changes to the repository will break the code. These conflicts in code stop others from being able to merge their code until the issues are resolved, so that more issues aren’t created.

Similarly, teams can create different workflows to take the hassle out of reviewing code or making sure it runs properly. Workflows are processes that created by a team to run a specific set of tasks when specified. An example of this would be to run the code whenever a push it made to the main branch of the repository.

All these tools combine to make GitHub a go to tool for software developers, whether it is an independent project or a team of hundreds GitHub’s features are essential for anyone who wants to take software developing seriously. Being able to go back un undo changes to code that were unintentional is a feature that is super helpful when it has been a while since you have looked at a specific part of a project.

In conclusion, GitHub is the best way to work on any software development projects. The ease of use and wide array of features make it the most effective tool for software developers. Whether it be your first-time coding or you’re a seasoned pro the different things that GitHub allows you to do with your code is amazing. Going back in time, being able to keep a detailed history of all changes made, and having the ability to make workflows that can look over and test the code that was written are all reasons why GitHub is so popular and will remain the premier software development tool.