**The three Stages of a File**

A file that resides in the committed stage means that the data in this file is safely stored in Steve’s local database with that project.

When Steve makes changes to his recipe file, the status of the file moves from committed, or unmodified, to modified.

This modified state means that the file has been altered from the last committed version that is stored in the local database.

Basically, this state just means that Steve has introduced new changes to the recipe file, but hasn’t committed them yet.

You can think of this state as a work in progress where more changes can be made to that file.

Now, when Steve has finished the changes he wants to make to a file, the file then moves to the staged state.

This means that Steve wants to take the changes made to that file during the modified state and then commit them to his local database to be in his next commit snapshot.

Now, It’s important to note that these three stages are applied to files in his project that are being tracked by Git, In other words, these stages only apply to files that were in the project’s last commit snapshot. A file can also be untracked.

Untracked files are any files added to the project since the last snapshot that Git hasn`t been tracking. To start tracking an untracked file Steve can simply add it to his Git project, and then Git places that file in the staged state, and it will be added into the next commit snapshot.

**The Three states of a Git Project**

Similar to how a file can reside in three different states, Steve’s Git project is made up of three main parts.

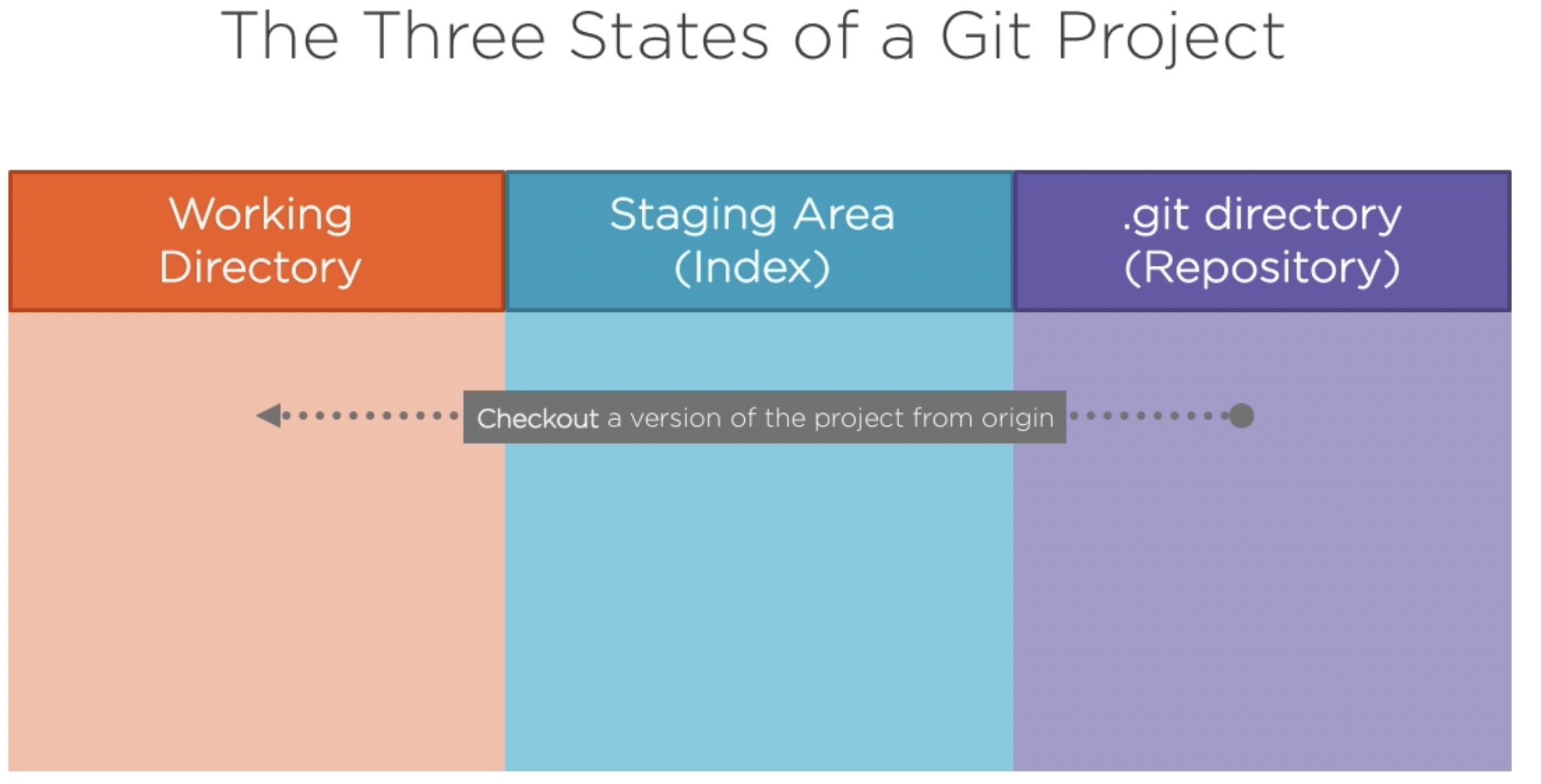
We start with the .git directory, also known as a repository. This is the origin of Steve’s project’s data and is what is pulled down from the remote server, which in Steve’s case is currently being stored on his local computer.

This is where Git stores the metadata and object database for his project.

Next we have the working directory.

This is a single copy, also known as a checkout of one version of the project.

These files are pulled from Steve’s local compressed database in the .git directory and placed in the working directory that he can then modify. (checkout a version of the project from origin)



The staging area is also known as the index. This is an area that sits between his working directory and the .git directory.

Steve can use the staging area to build up a change or a set of changes that he wants to commit by taking a snapshot of his project with those changes in place, and when he creates a commit.

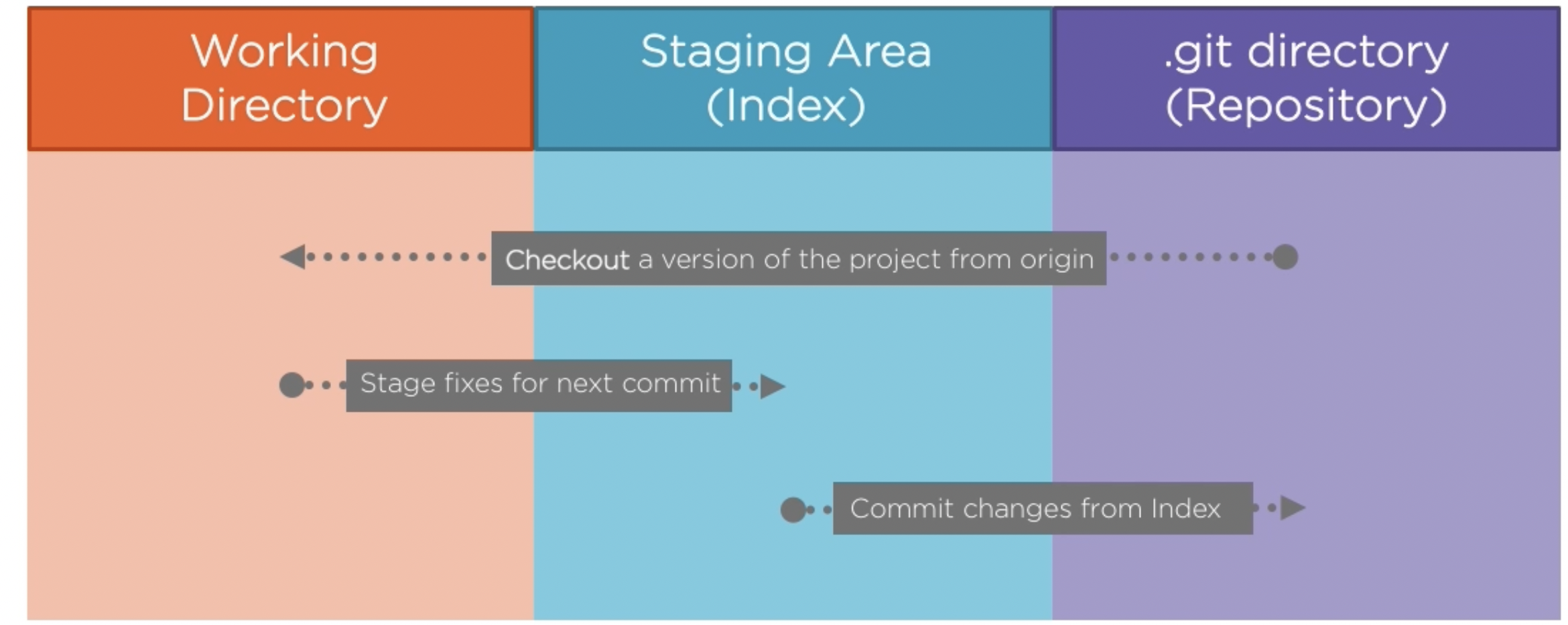


What is committed is only what is currently in the staging area, not what is in his working directory.

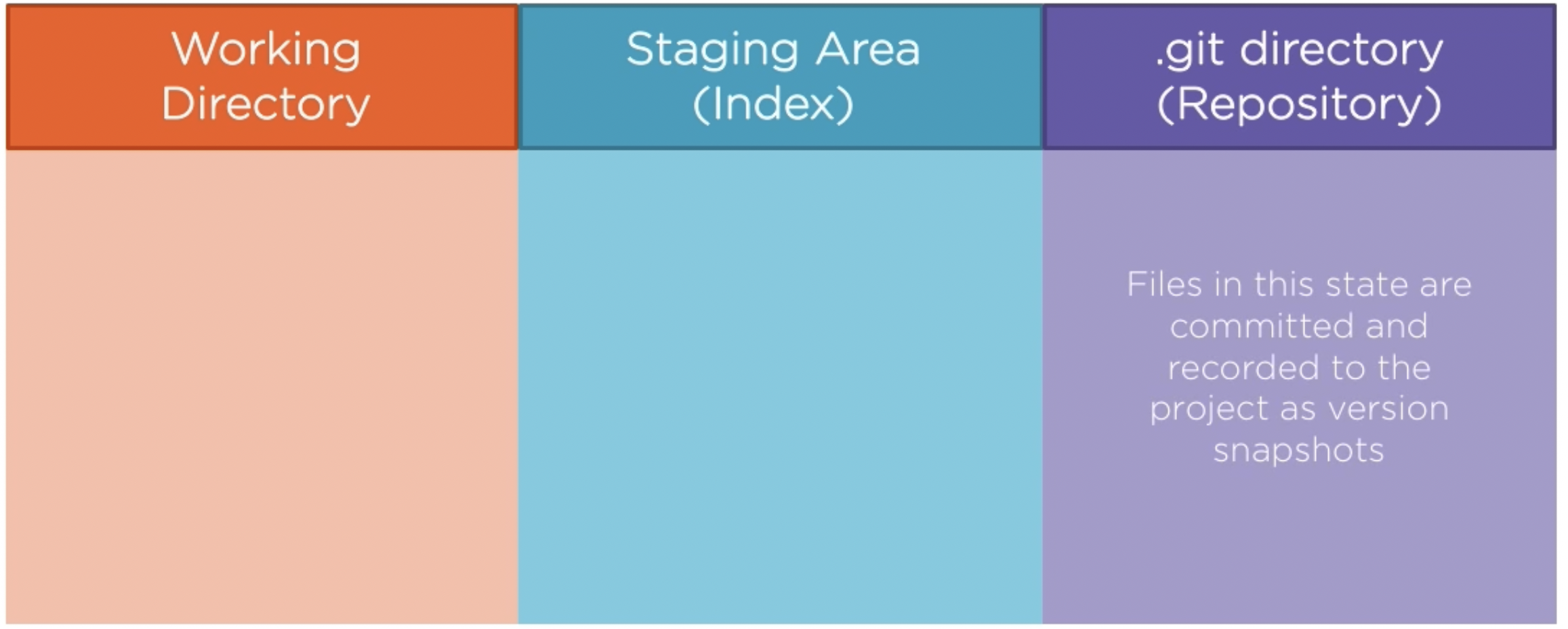
This gives Steve the freedom to modify several files, but only commit the changes of a file or files that he wants to.

The uncommitted changes remain in his working directory until Steve decides to commit them or stash them away for future use.

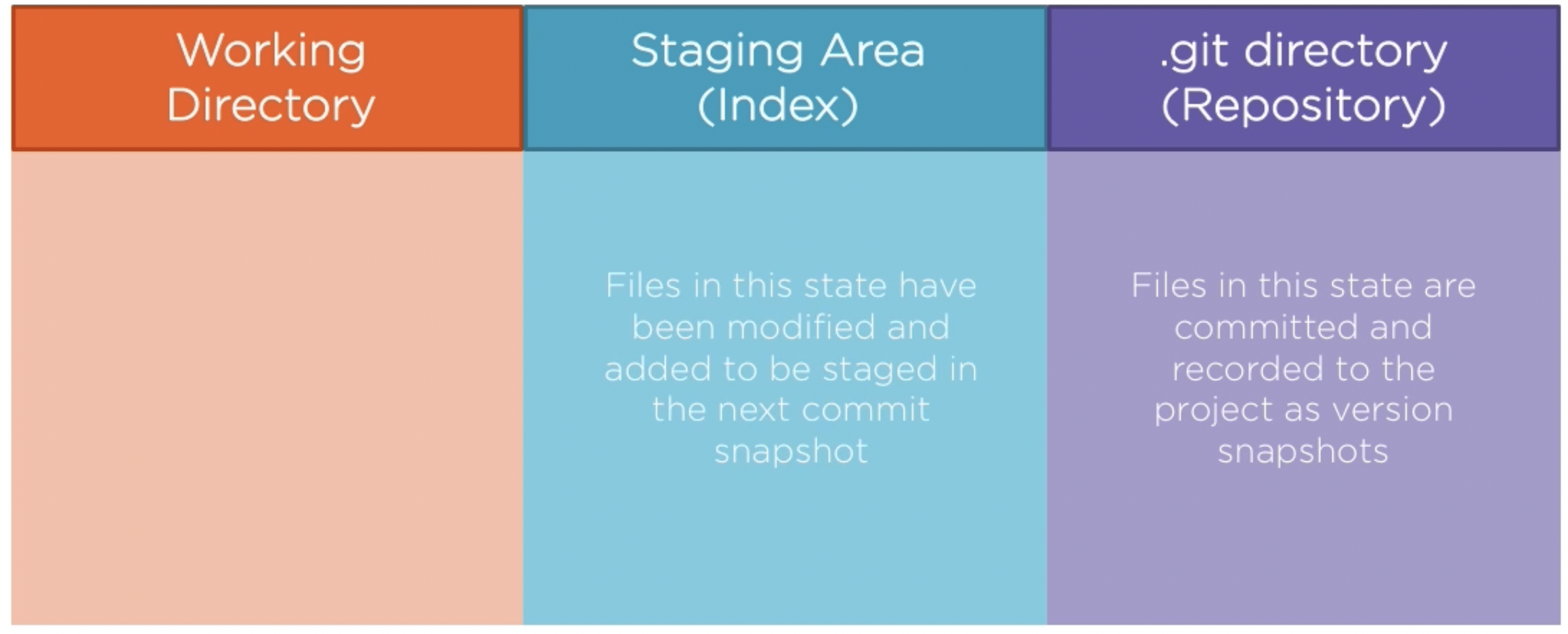
After Steve commits his changes, he then pushes his commits up to the .git directory to update the origin data of his project.



So, in short, if a file version is in the .git directory, it’s committed and recorded in that project’s repository;



if a file is modified and has a been added to the staging area, it’s staged;



And if a file was modified since the last checkout of that file from the .git dirfectory, but has not been added to the staging area, it’s modiofied.

