Blog

How to Delete Commit History – A Step-by-Step Guide

Jochem Loedeman 09 Mar, 2023

GitHub

Share

While developing software, commit history holds all the information about changes made to that project. There are certain situations where you need to erase commit history to make your repository cleaner and safer. In this blog, we are going to see how to delete commit history safely. We will make space to start afresh with these steps.

At Xebia, we promote publishing software projects as a way to give back to the community, promote innovation, and demonstrate authority. But when doing so, it is important to consider the commit history of the project.

The **commit history** is a record of all the changes made to the project over time. This includes code changes, bug fixes, and other modifications. Sometimes it may be desirable to delete the commit history of a project before publishing it. For example, your project may have had a very different purpose and scope in the past, making the commit history irrelevant and confusing for anyone who was not involved in the development.

In this blog post, we will walk you through the steps to delete the commit history of a code project so that it is ready to publish!

Make sure your main branch is in the desired state first, as it will become the starting point of your new history. Then run

```
git checkout --orphan temp_branch
```

This command creates a new branch named temp_branch and switches to it. The `--orphan` option creates a branch without a commit history.

Step 2: Add all files

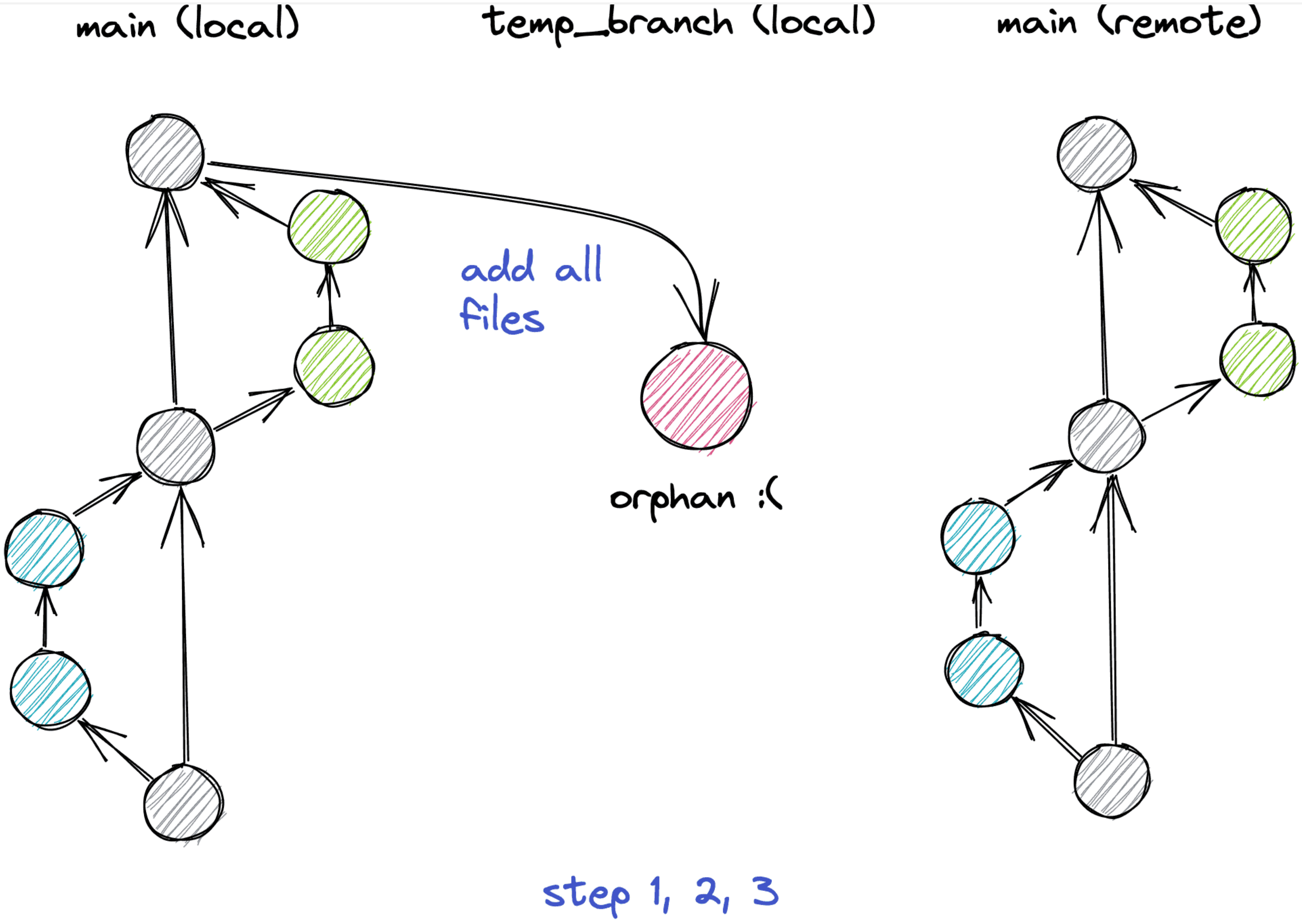
```
git add -A
```

This will stage all changes in the working directory, including deleted files.

Step 3: Commit the changes to the commit history

We can now commit the changes to the orphan branch using the following command:

```
git commit -m "Initial commit"
```


main (local)
temp_branch (local)
main (remote)
add all files
orphan :(
step 1, 2, 3

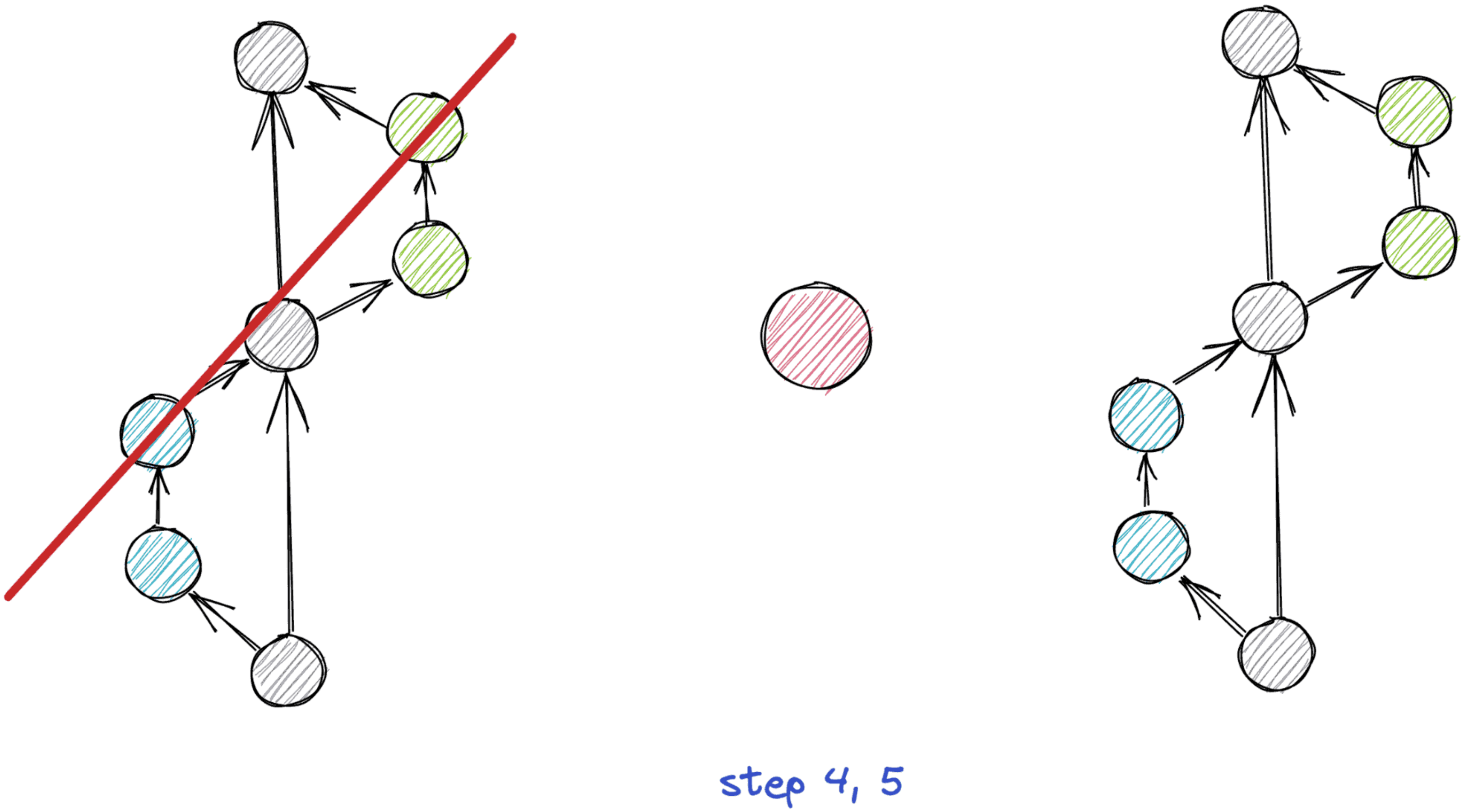
Now that we have committed the changes to the new branch, we can delete the local main branch, which contains our full commit history.

```
git branch -D main
```

Step 5: Rename the temporary branch to main

Next, we need to rename the temporary branch to master:

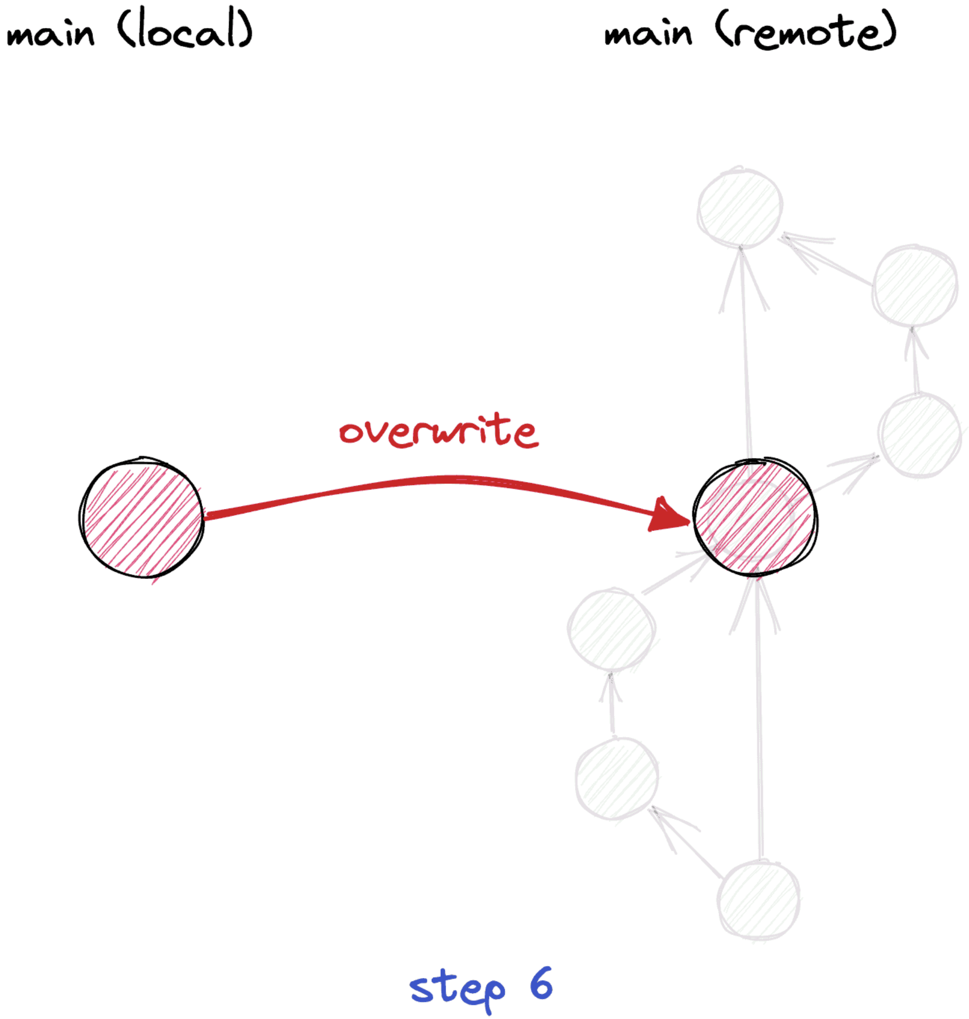
```
git branch -m main
```



Step 6: Force update to our Git repository

Finally, we need to force update our git repository using the following command:

The --force option allows us to overwrite the history of the remote main branch with our new local history. This option should generally be used with caution, because you risk losing work you or others committed in the past. However, for us this is the intended result

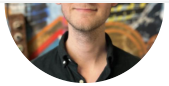


Deleting the commit history of a project before publishing it on GitHub is a serious decision that should not be taken lightly. After all, you will lose all changes ever made to the project, which makes it impossible to revert to an earlier version of your codebase. It is important to consider the reasons why you want to delete the commit history and the potential impact it may have on other contributors to the project. However, if you do decide to delete the commit history, the steps outlined above can help you do it effectively and efficiently.

Sources –

GitHub – Delete commits history with git commands · GitHub

‹packt›
GitHub Actions
Cookbook
A practical guide to automating repetitive tasks and streamlining your development process
MICHAEL KAUFMANN



Questions?

Get in touch with us to learn more about the subject and related solutions

Contact Us

Legal

Creating Digital Leaders.

Connect