Statistical Programming and Open Science Methods

Using Git and Github

Joachim Gassen
Humboldt-Universität zu Berlin

September 02, 2022

SFB/Transregio 266
ACCOUNTING FOR TRANSPARENCY

Time table October 10

When?	What?
09:00	Welcome and Introduction
09:30	The development environment and project organization
10:30	Coffee
11:00	Using Git and Github
12:30	Lunch
14:00	Statistical programming languages: An overview
15:30	Coffee
16:00	Functional versus object-oriented programming
19:30	Pizza at Due Forni, Schönhauser Allee 12

Disclaimer

Some of the following, in particular the figures, are borrowed from:

Scott Chacon and Ben Straub (2014): Pro Git,
https://git-scm.com/book/en/v2

The issue

We all know these folders

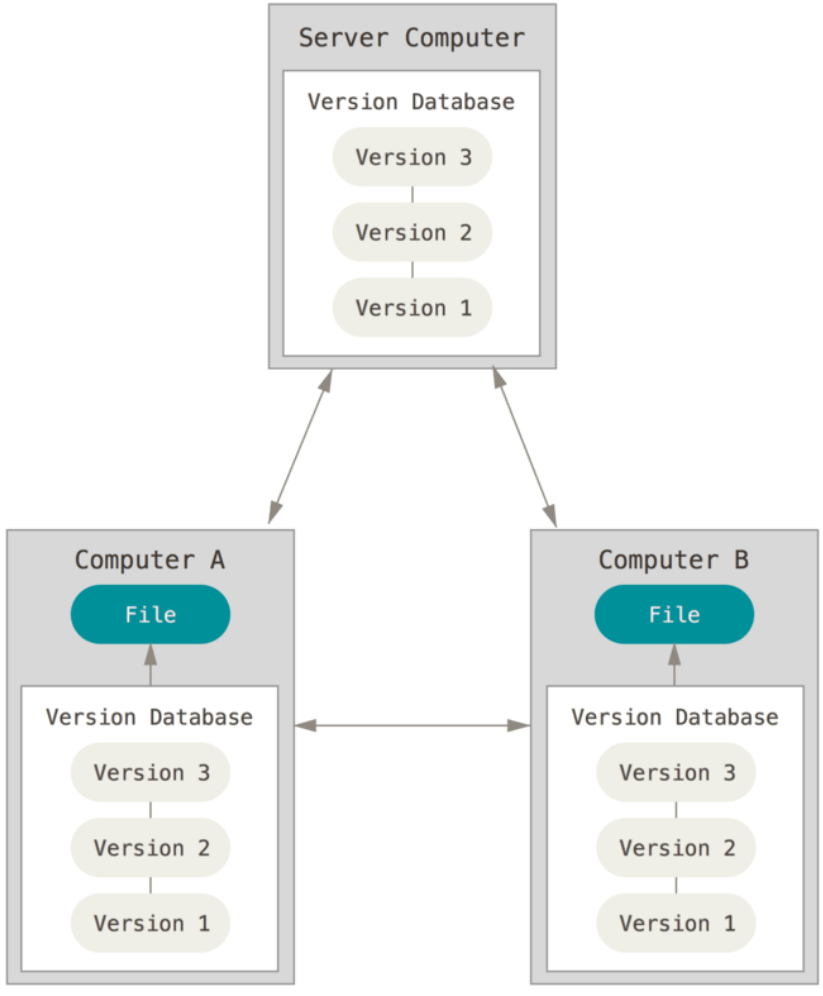
```
joachim:project/paper$ ls

paper_draft4_2019-08-17a.tex
paper_draft4_2019-08-17a_jg.tex
paper_draft3_dv_no_track_changes.tex
thoughts-on-draft.txt
paper_draft4_current.tex
old_stuff/
```

The solution

The same folder with git:

```
joachim:project/paper$ ls -a
.
..
.git/
paper.tex
```

Git is an open source distributed version control system



Let's start

TASK: Write code to import the data of the latest ZIP file from https://www.sec.gov/dera/data/financial-statement-data-sets.html

Your first commit

- Tell git who you are

```
joachim:$ git config --global user.name "Joachim Gassen"
joachim:$ git config --global user.email "gassen@wiwi.hu-berlin.de"
```

- Clone our project from Github

```
joachim:$ git clone https://github.com/joachim-gassen/sposm
```

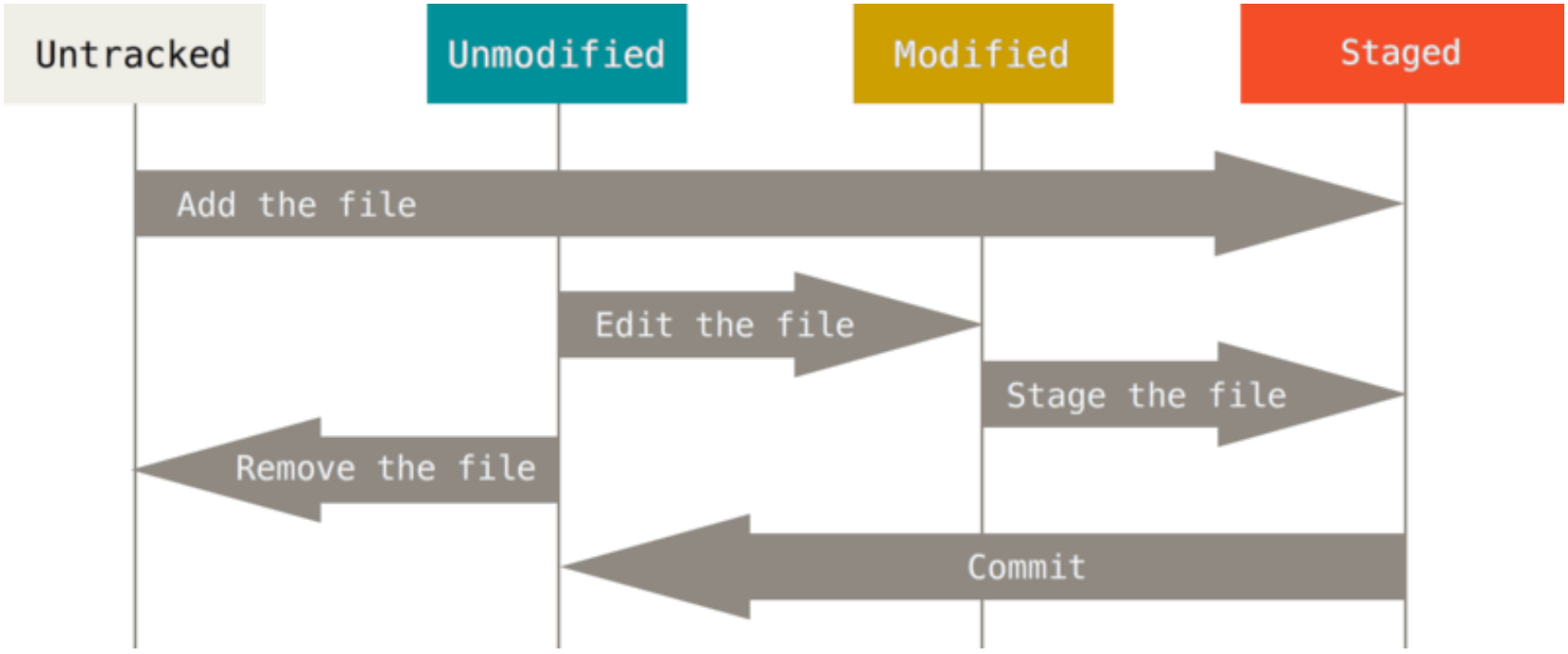
- Create a new code file in the code directory and add it to git (stage it)

```
joachim:$ cd sposm/code
joachim:sposm/code$ nano read_sec_fin_stat_data.R
joachim:sposm/code$ git add read_sec_fin_stat_data.R
```

- Then commit your changes to your local repository

```
joachim:sposm/code$ git commit -m "Started on SEC import code"
```

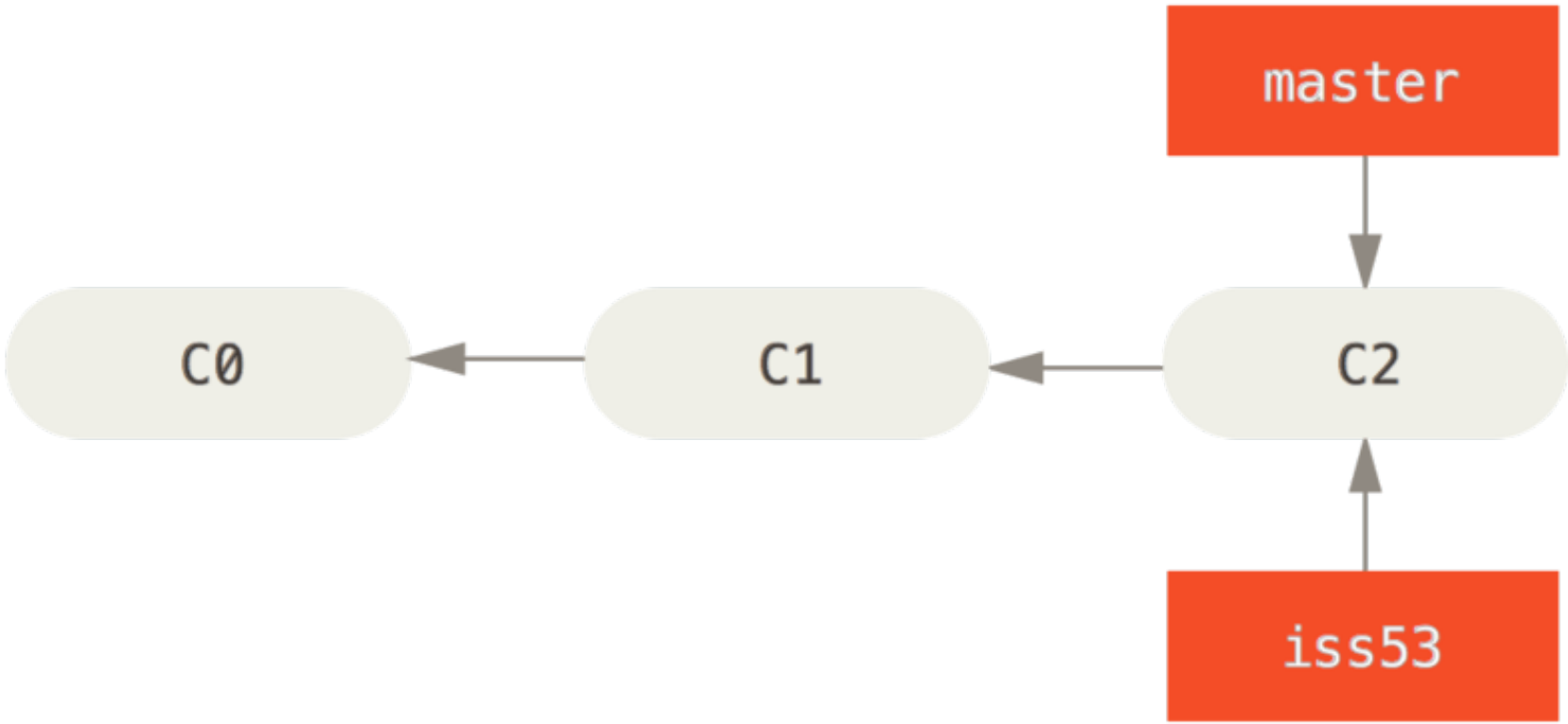
What have you just done?



Time to branch

- Create a new branch and change to that branch

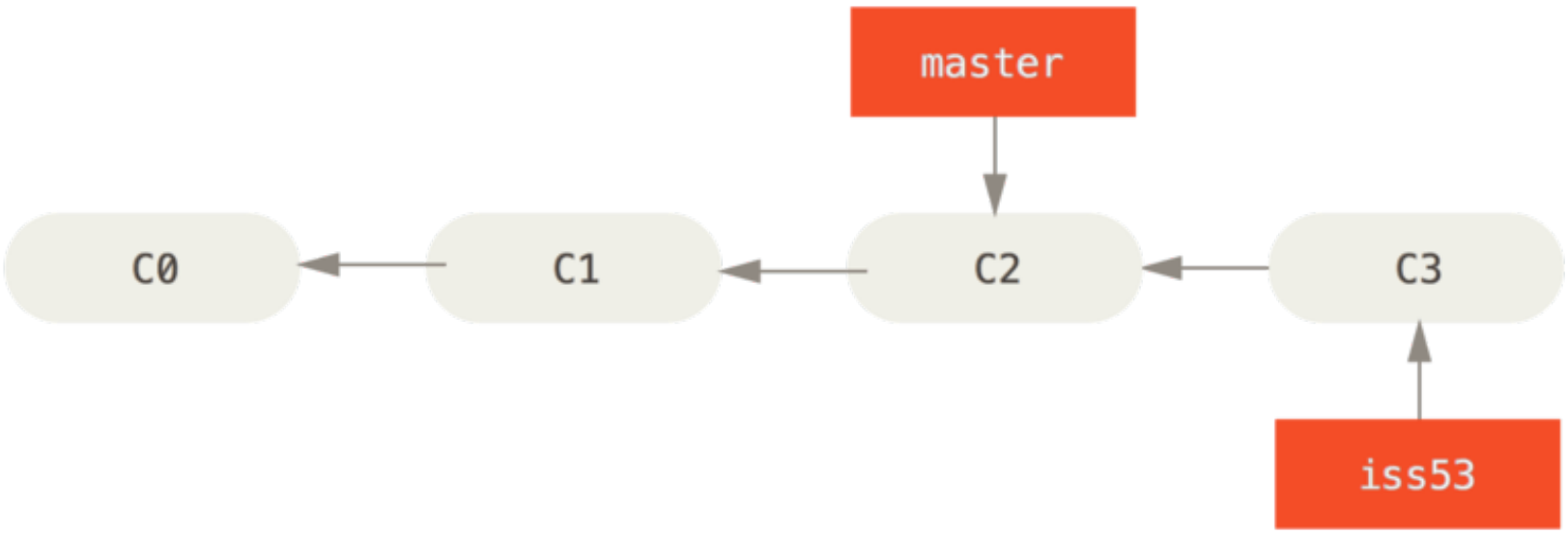
```
joachim:sposm/code$ git branch iss53
joachim:sposm/code$ git checkout iss53
```



Make some commits

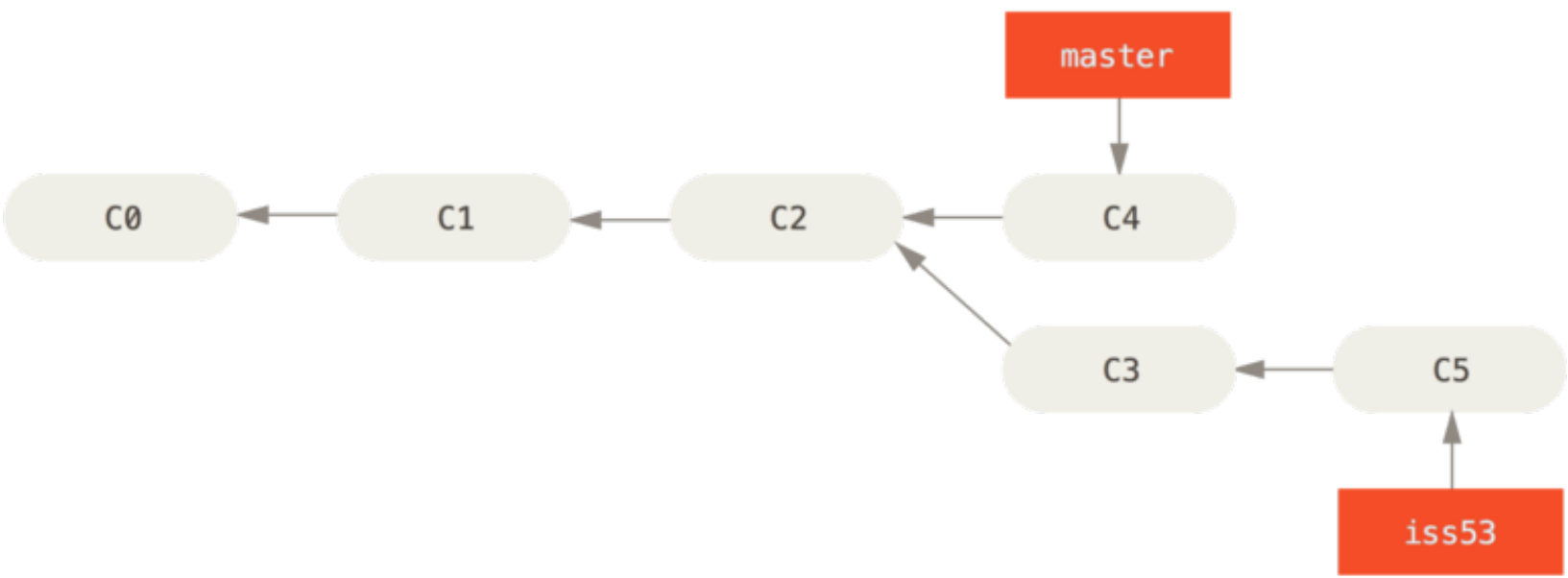
- Make commits to branch

```
joachim:sposm/code$ nano read_sec_fin_stat_data.R
joachim:sposm/code$ git add read_sec_fin_stat_data.R
joachim:sposm/code$ git commit -m "Some new stuff on iss53"
```



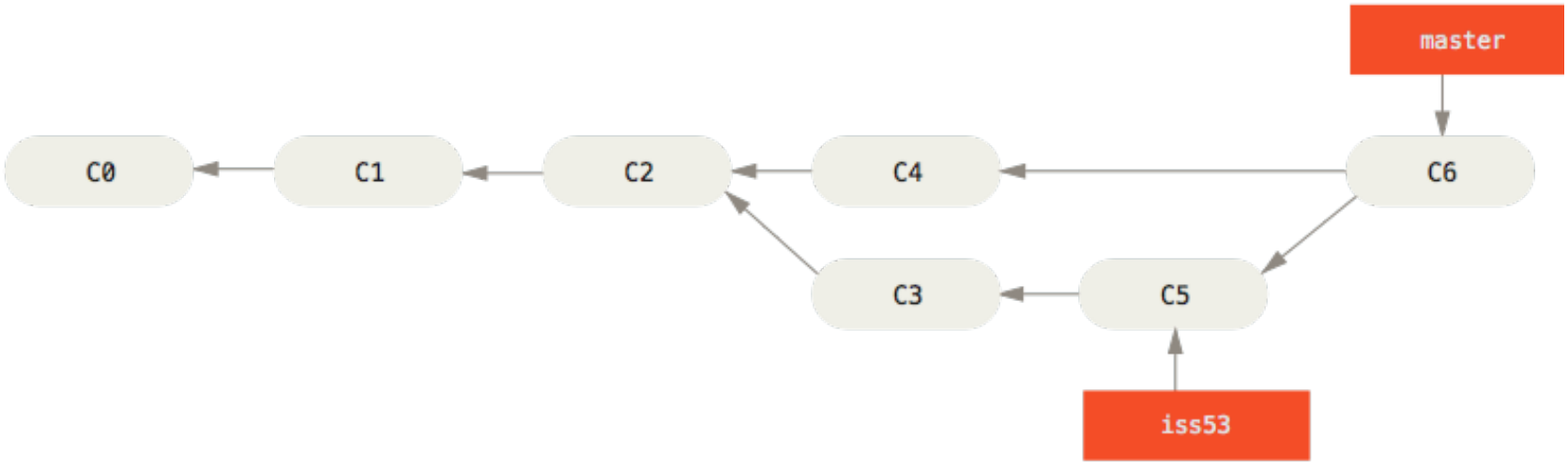
And continue (on both branches)

```
joachim:sposm/code$ git checkout master
joachim:sposm/code$ nano read_sec_fin_stat_data.R
joachim:sposm/code$ git add read_sec_fin_stat_data.R
joachim:sposm/code$ git commit -m "Some urgent hotfix"
joachim:sposm/code$ git checkout iss53
joachim:sposm/code$ nano read_sec_fin_stat_data.R
joachim:sposm/code$ git add read_sec_fin_stat_data.R
joachim:sposm/code$ git commit -m "Even more new stuff on iss53"
```



And now? Two Options: 1. Merge

```
joachim:sposm/code$ git checkout master
joachim:sposm/code$ git merge iss53
```




```
# delete the old branch
joachim:sposm/code$ git branch -d iss53
```

This does not always work directly

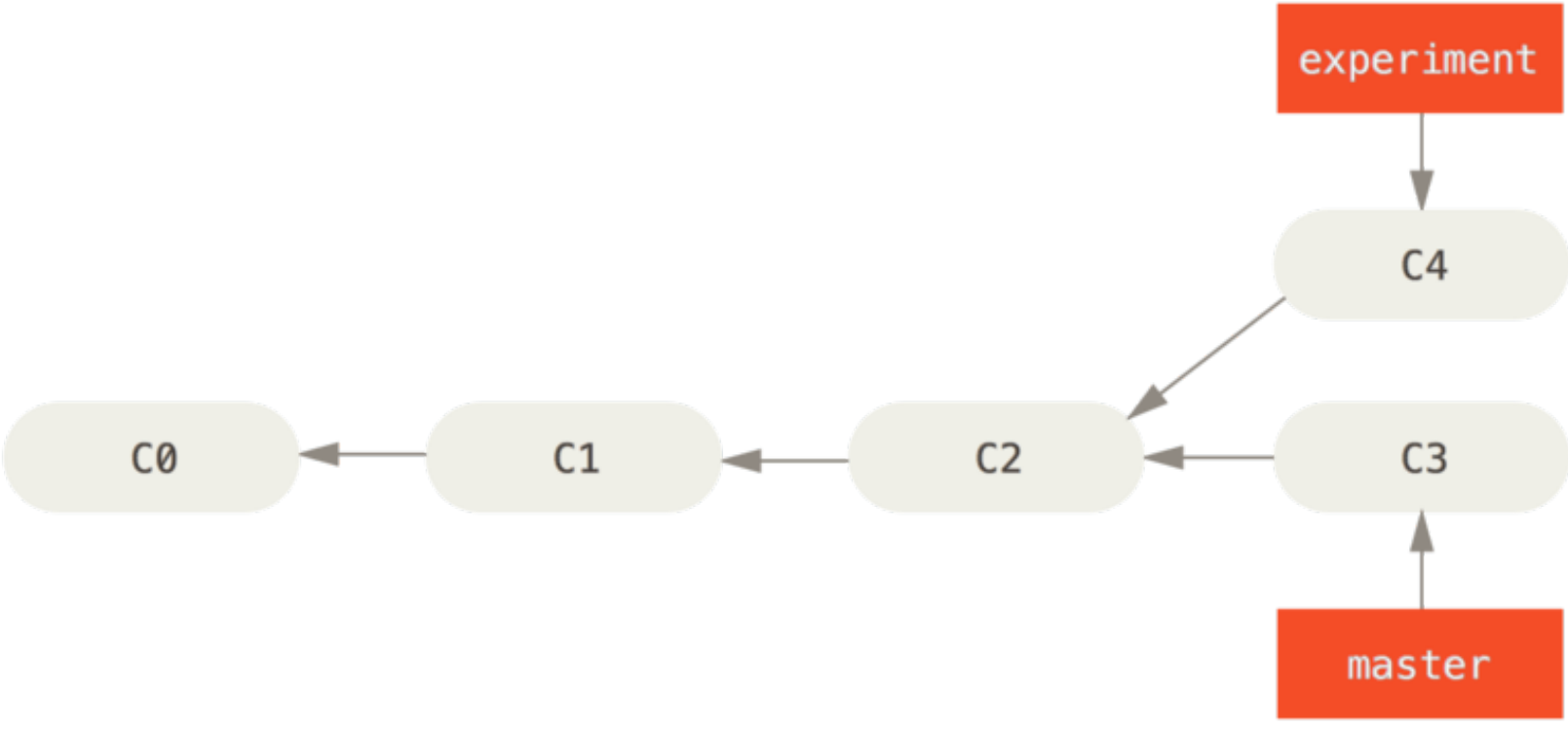
```
joachim:sposm/code$ git merge iss53
Auto-merging code/read_sec_fin_stat_data.R
CONFLICT (content): Merge conflict in code/read_sec_fin_stat_data.R
Automatic merge failed; fix conflicts and then commit the result.
joachim:sposm/code$ nano read_sec_fin_stat_data.R
```

- You actually have to work now

```
joachim:sposm/code$ nano read_sec_fin_stat_data.R

# Editing file to remove conflicts (not always trivial)

joachim:sposm/code$ git add read_sec_fin_stat_data.R
joachim:sposm/code$ git commit -m "Merged iss53 into master"
joachim:sposm/code$ git branch -d iss53
```

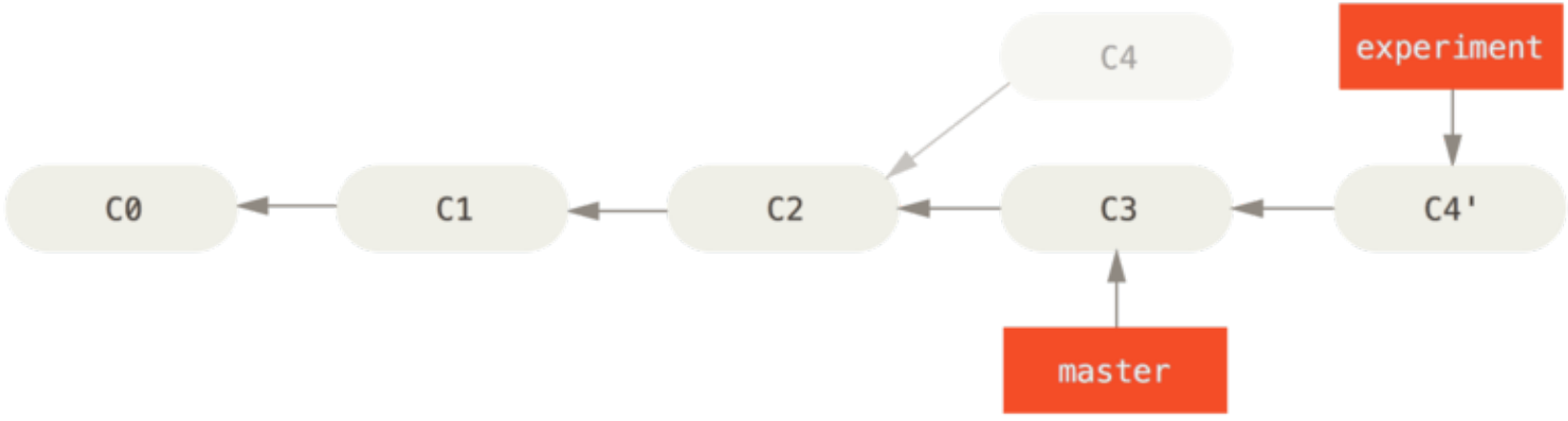
The second option: Rebase

- Assume that you generated a new branch `experiment` and have reached the state below



Rebasing . . .

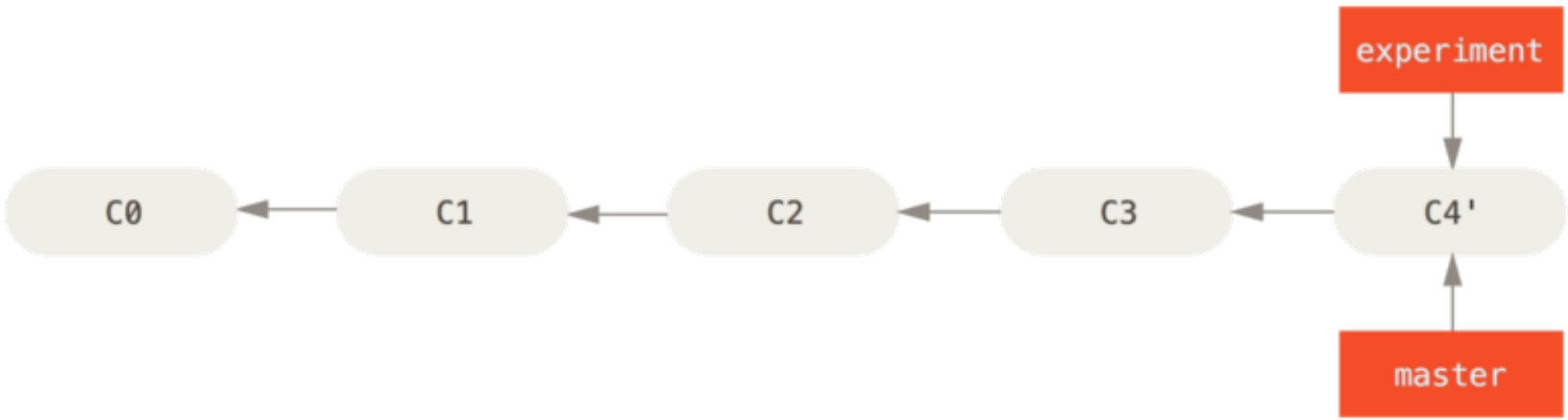
```
joachim:sposm/code$ git checkout experiment
joachim:sposm/code$ git rebase master
```



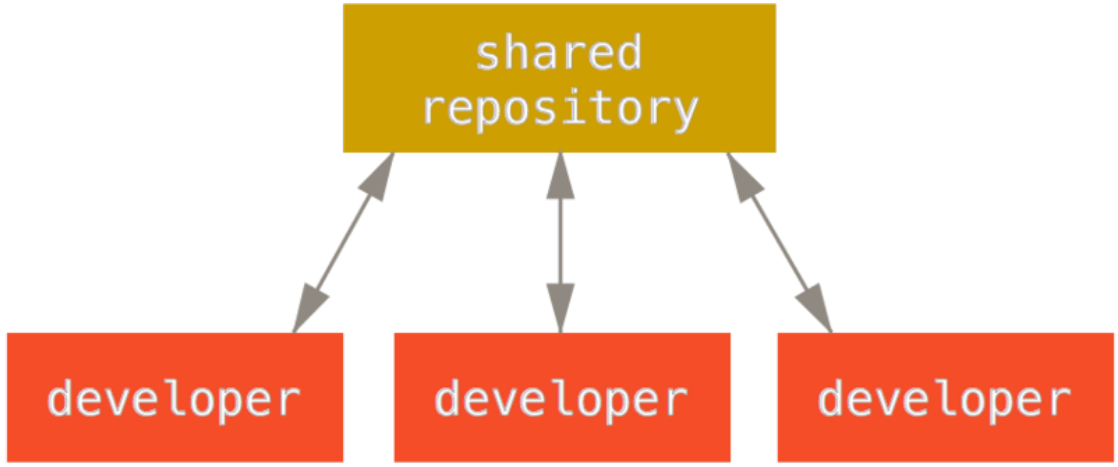
- This can create conflicts just like merging

. . . and fast-forward merging

```
joachim:sposm/code$ git checkout master
joachim:sposm/code$ git merge experiment
joachim:sposm/code$ git branch -d experiment
```



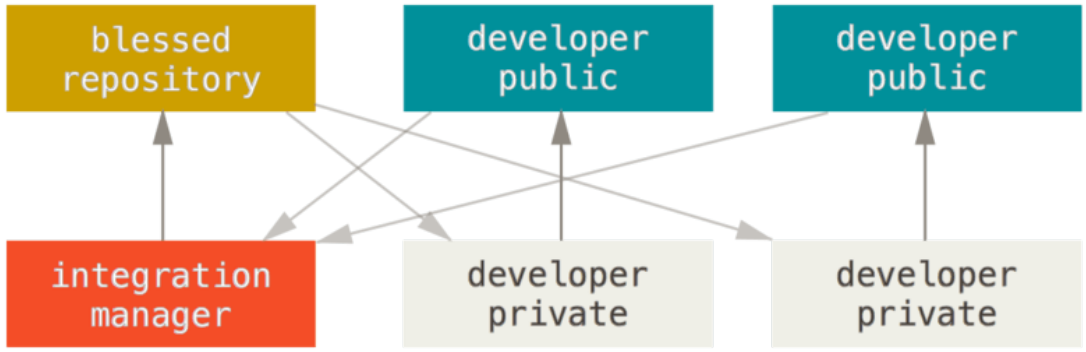
Using remote repositories: Collaborative approach



- All developers push to the same remote repository. Can be fast and efficient but requires that people know what they are doing as there is no central oversight

```
joachim:sposm/code$ git remote -v
origin  https://github.com/joachim-gassen/sposm (fetch)
origin  https://github.com/joachim-gassen/sposm (push)
joachim:sposm/code$ git pull
joachim:sposm/code$ git push
```

Using remote repositories: Fork & pull request approach



- Developers push their own forks of a central repository. Maintainer of central repository decides on pull requests from these remote repositories. Pull requests are communicated via Github (or by other means)
- For more details see: https://gist.github.com/Chaser324/ce0505fbed06b947d962

```
# Fork repository on github and clone your local fork
joachim:sposm$ git remote add up https://github.com/joachim-gassen/sposm
joachim:sposm$ git fetch up
joachim:sposm$ git checkout master
joachim:sposm$ git merge up/master
joachim:sposm$ git push
# Issue pull request to get your stuff into the central repository
```

Additional stuff that Github brings to the party

- Besides providing publicly hosted repositories Github provides several features that make collaboration easier
- Web interface for exploring repositories, making commits and issuing pull requests
- Issue tracking system
- Website hosting for code documentation
- ...

Assignment for the break

Use the 2019Q2 SEC data to extract current quarterly revenues of U.S. based firms. Try to replicate the following methodology.

We use the SEC Financial Statement Dataset to obtain the most current quarterly revenues of U.S. based SEC registrants. To distill total revenue from reported XBRL data, we take for each filing the maximum value of the three tags

- `"Revenues"`
- `"RevenueFromContractWithCustomerExcludingAssessedTax"`
- `"RevenueFromContractWithCustomerIncludingAssessedTax"`

Produce a clean sample and report the sample size, the sample's mean and the sample's standard deviation.

For extra credit: Prepare a map that shows where in the U.S. the corporate revenues are located.