

# **GIT**

## **THE SVN OF MODERN TIMES**

liviu.iosif@[code932.com](mailto:code932.com)

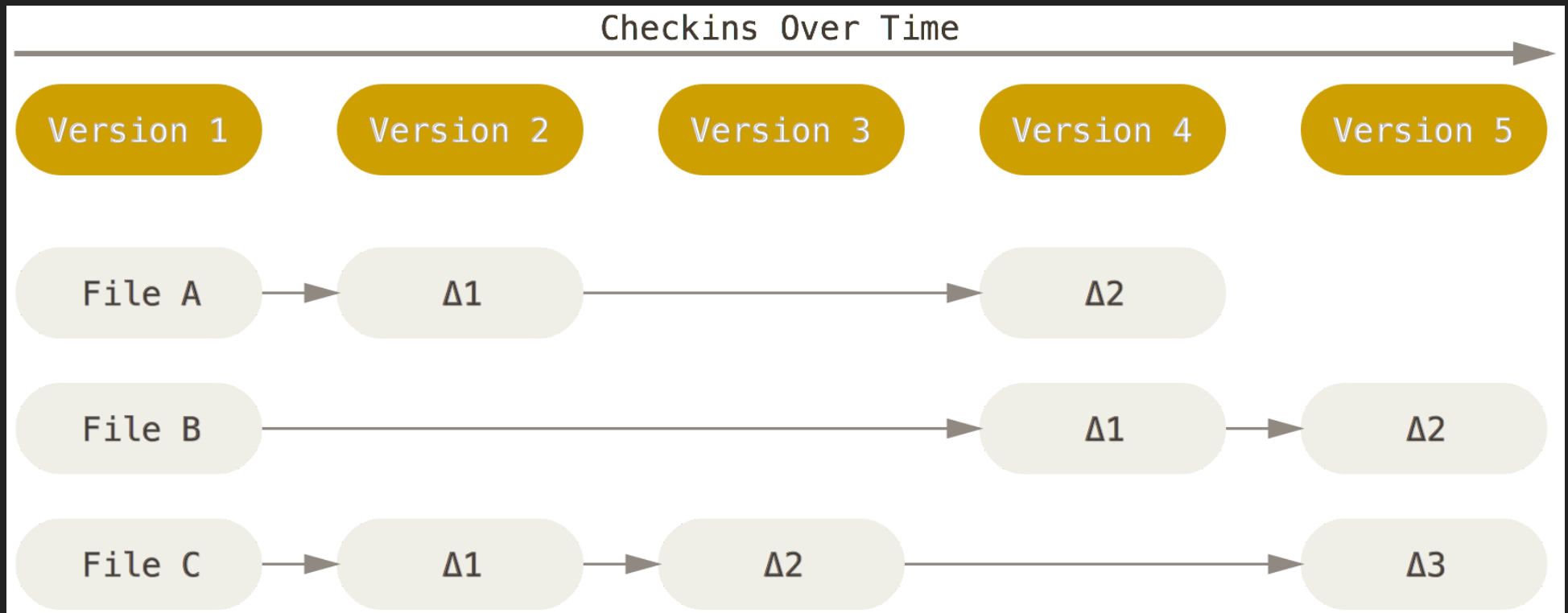
# AGENDA

1. A short history
2. How does it work
3. Up and running
4. Workflows
5. Going with the flow
6. Branches
7. Merges and patches
8. Password caching
9. Best practices

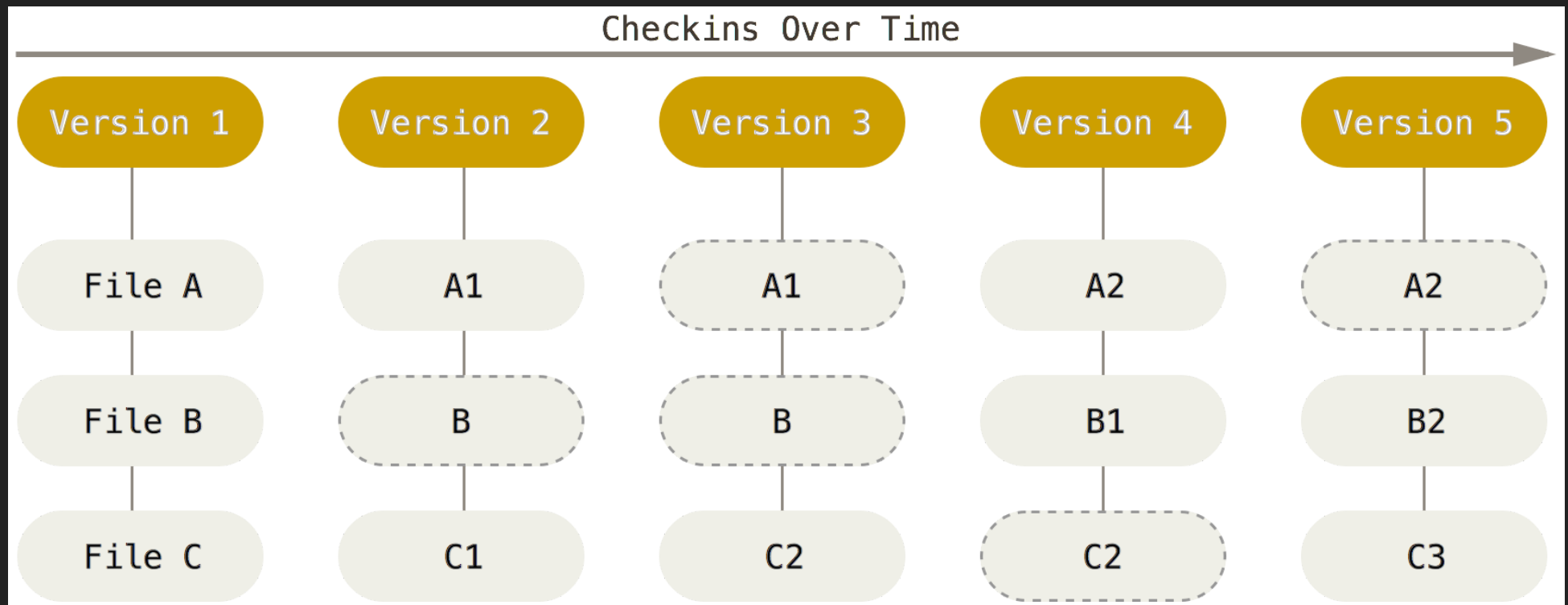
# A REALLY SHORT HISTORY OF GIT

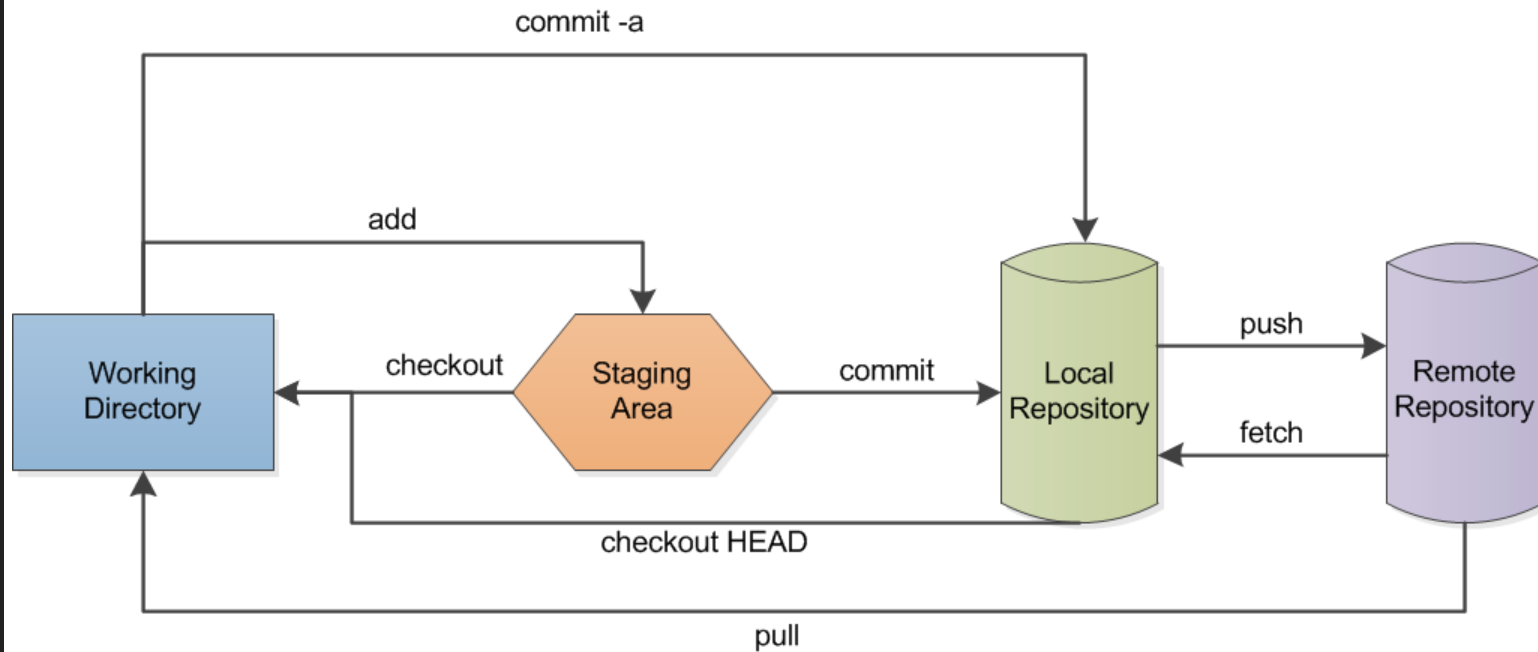
**HOW DOES IT WORK**

# DIFFERENCES (SVN)



# SNAPSHOTS (GIT)





**commit -a**: Directly commit modified and deleted files into the local repository (*no new files!*)

**add**: Add a file to the staging area.

**checkout**: Get a file from the staging area.

**checkout HEAD**: Get a file from the local repository

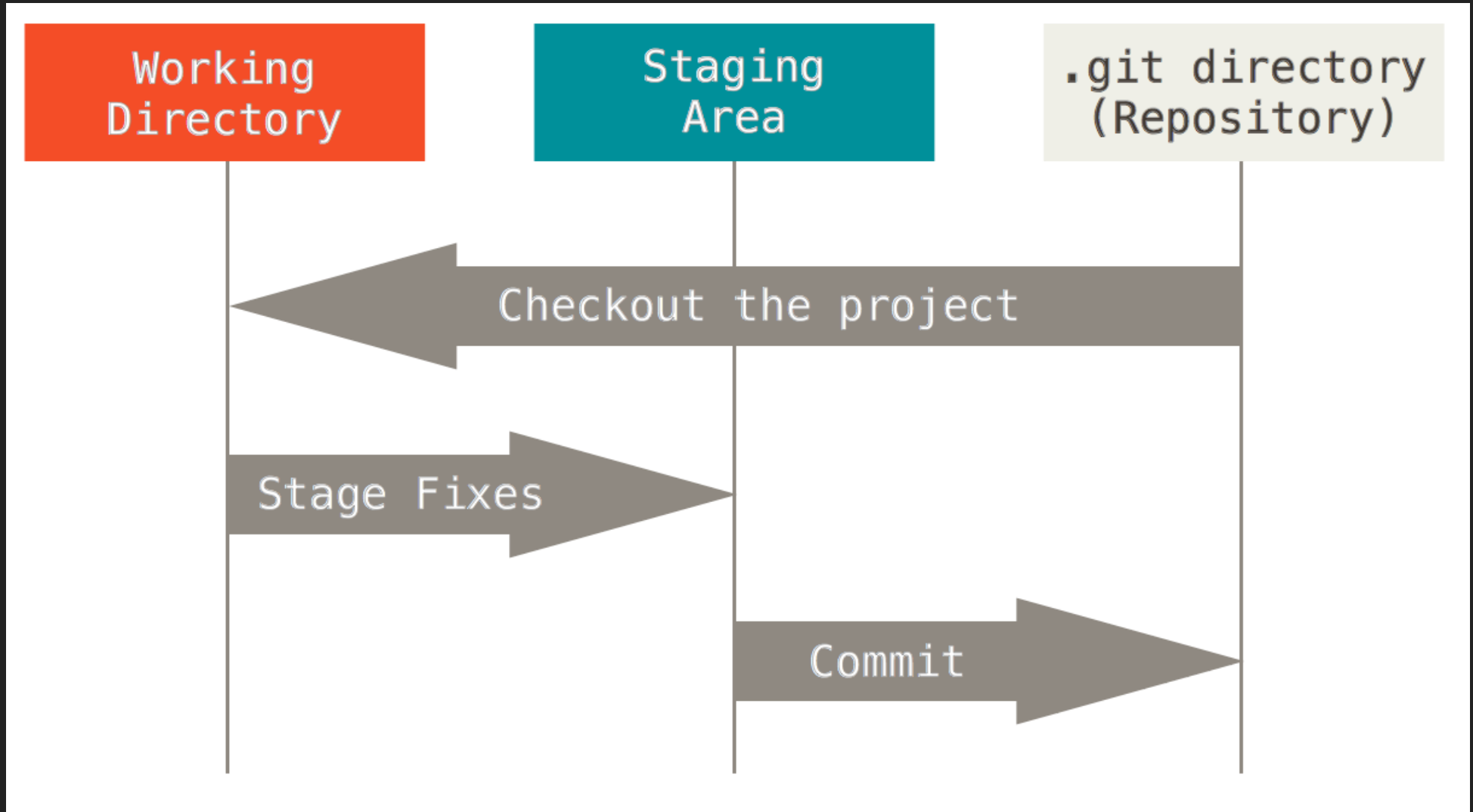
**commit**: Commit files from the staging area to the local repository

**push**: Send files to the remote repository

**fetch**: Get files from the remote repository

**pull**: Get files from the remote repository and put a copy in the working directory

# THE 3 STAGES





**UP AND RUNNING**

# INSTALL

## WINDOWS

Download from [here](#) and install

## LINUX

```
$ apt-get install git-all  
$ yum install git-all
```

## MAC

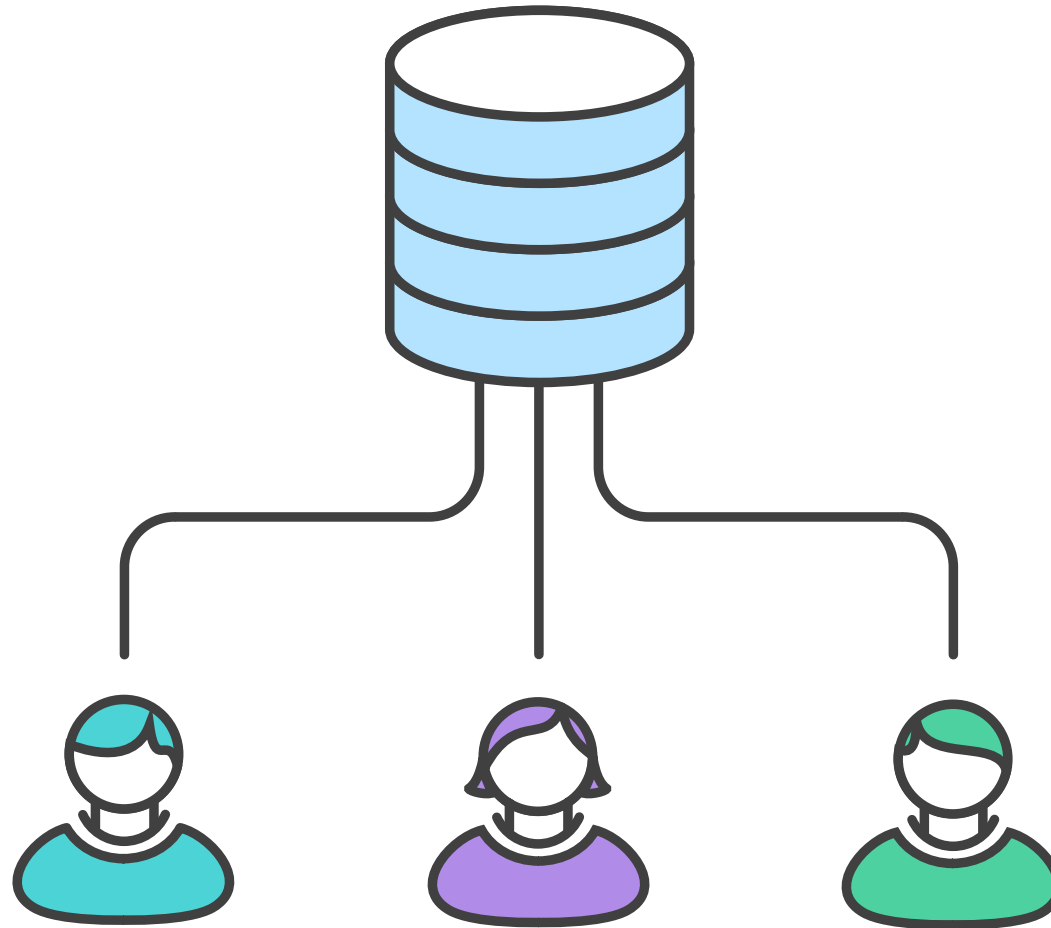
```
$ xcode-select --install
```

# CONFIG

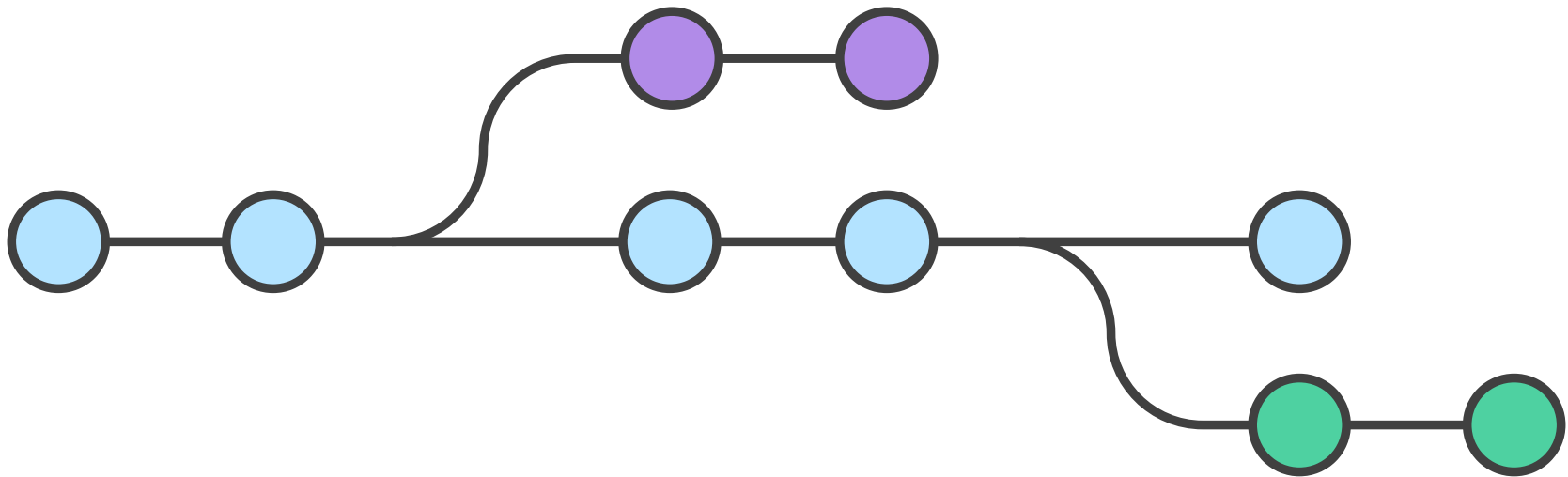
```
$ git config --local user.name "John Doe"  
$ git config --local user.email "john@doe.com"
```

# WORKFLOW GUIDELINES

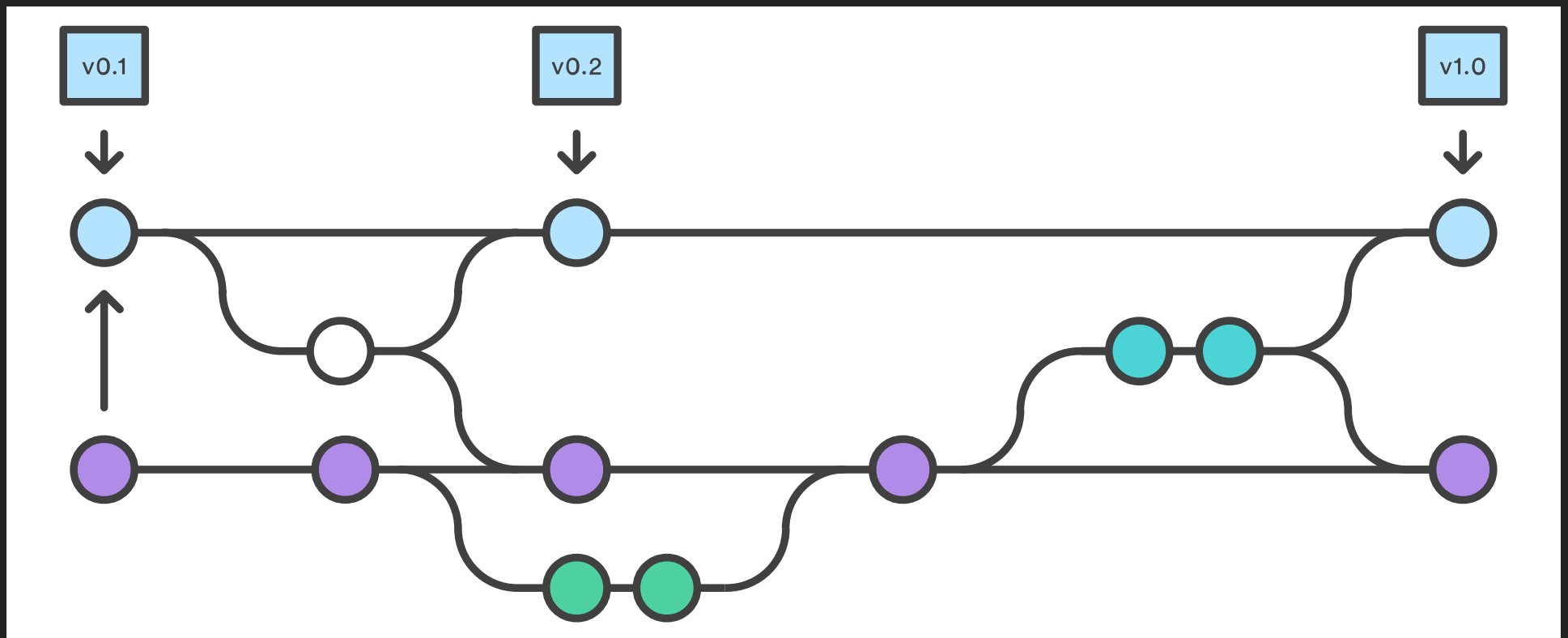
# CENTRALIZED WORKFLOW



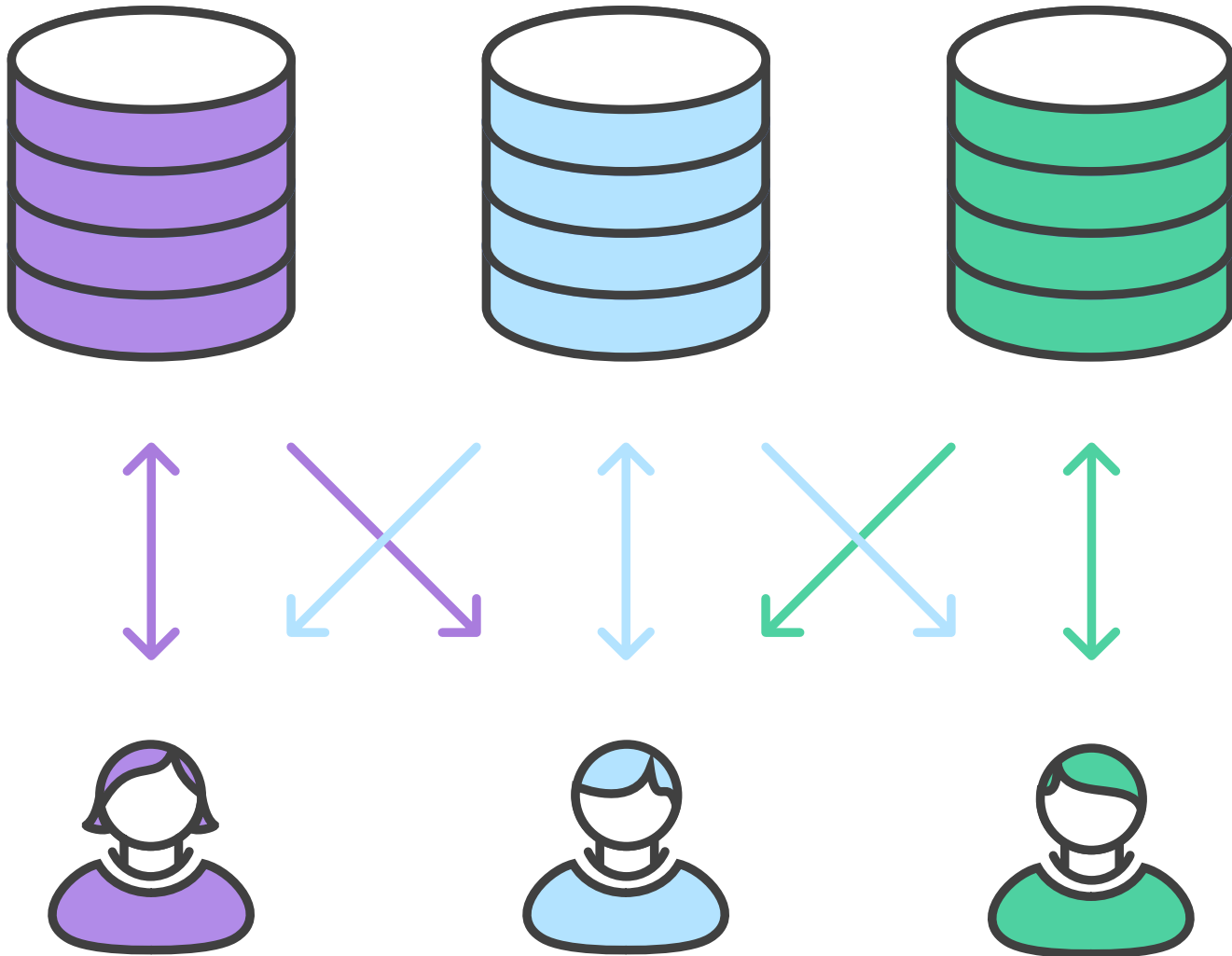
## FEATURE BRANCH WORKFLOW



# GITFLOW WORKFLOW



# FORKING WORKFLOW





**GOING WITH THE FLOW**

# A FEW COMMANDS

```
$ git help # shows help
$ git clone # clones a central repository into a local one
$ git status # shows the current status for changed/new/removed file
$ git add # stages file(s) for commit
$ git commit
$ git push # publishes the changes(commits) to the central repository
$ git pull # fetches changes from central repository
$ git branch # shows a list of existing branches
$ git log # shows history
$ git merge # merges 2 branches together
$ git stash # stacks unfinished changes
```



```
$ git status
On branch master
Your branch is up-to-date with 'origin/master'.
  (use "git push" to publish your local commits)
Changes not staged for commit:
      modified:   index.html
Untracked files:
  (use "git add <file>..." to include in what will be
      demo/

no changes added to commit (use "git add" and/or "git
```



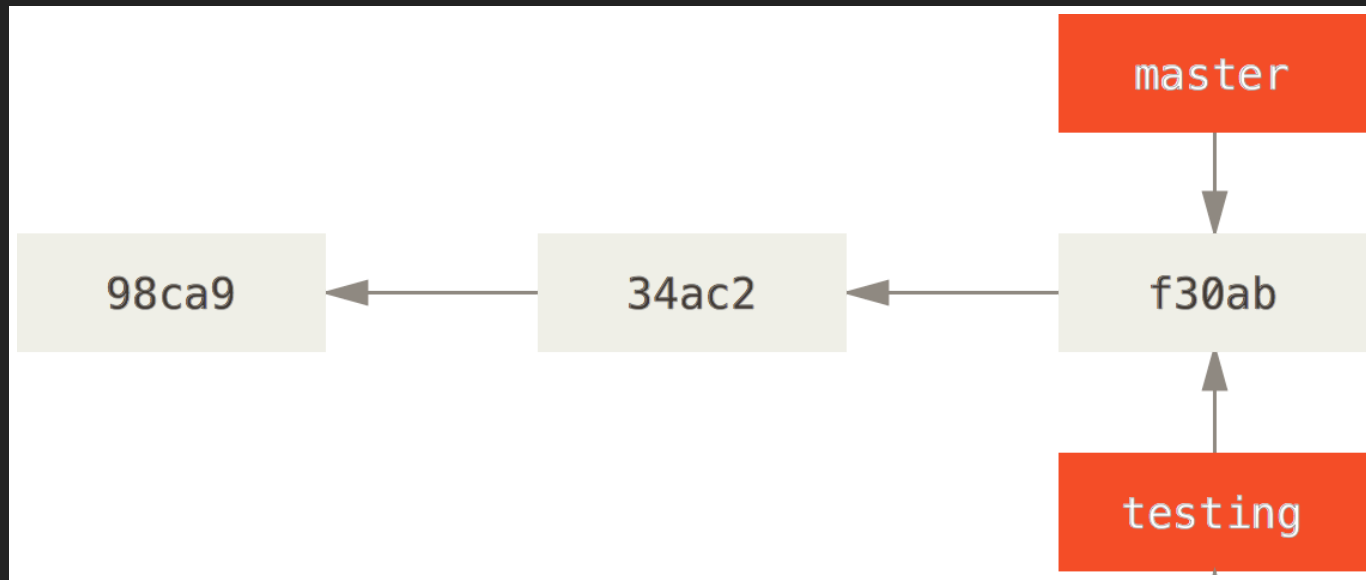


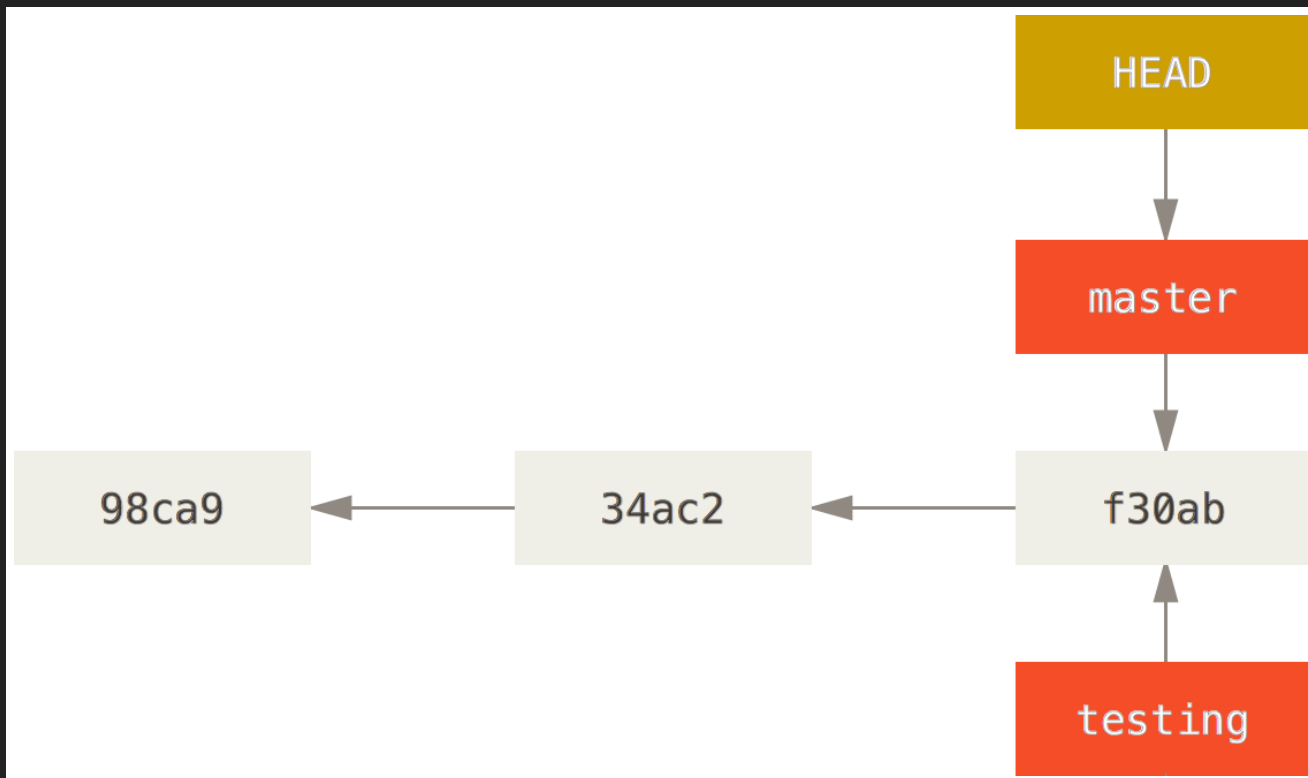
```
$ git push origin master
Password for 'https://iionut@github.com':
Counting objects: 4, done.
Delta compression using up to 8 threads.
Compressing objects: 100% (2/2), done.
Writing objects: 100% (4/4), 351 bytes | 0 bytes/s, do
Total 4 (delta 1), reused 0 (delta 0)
To https://github.com/iionut/git-presentation.git
8fba580..db0d959  master -> master
```

**BRANCHES**

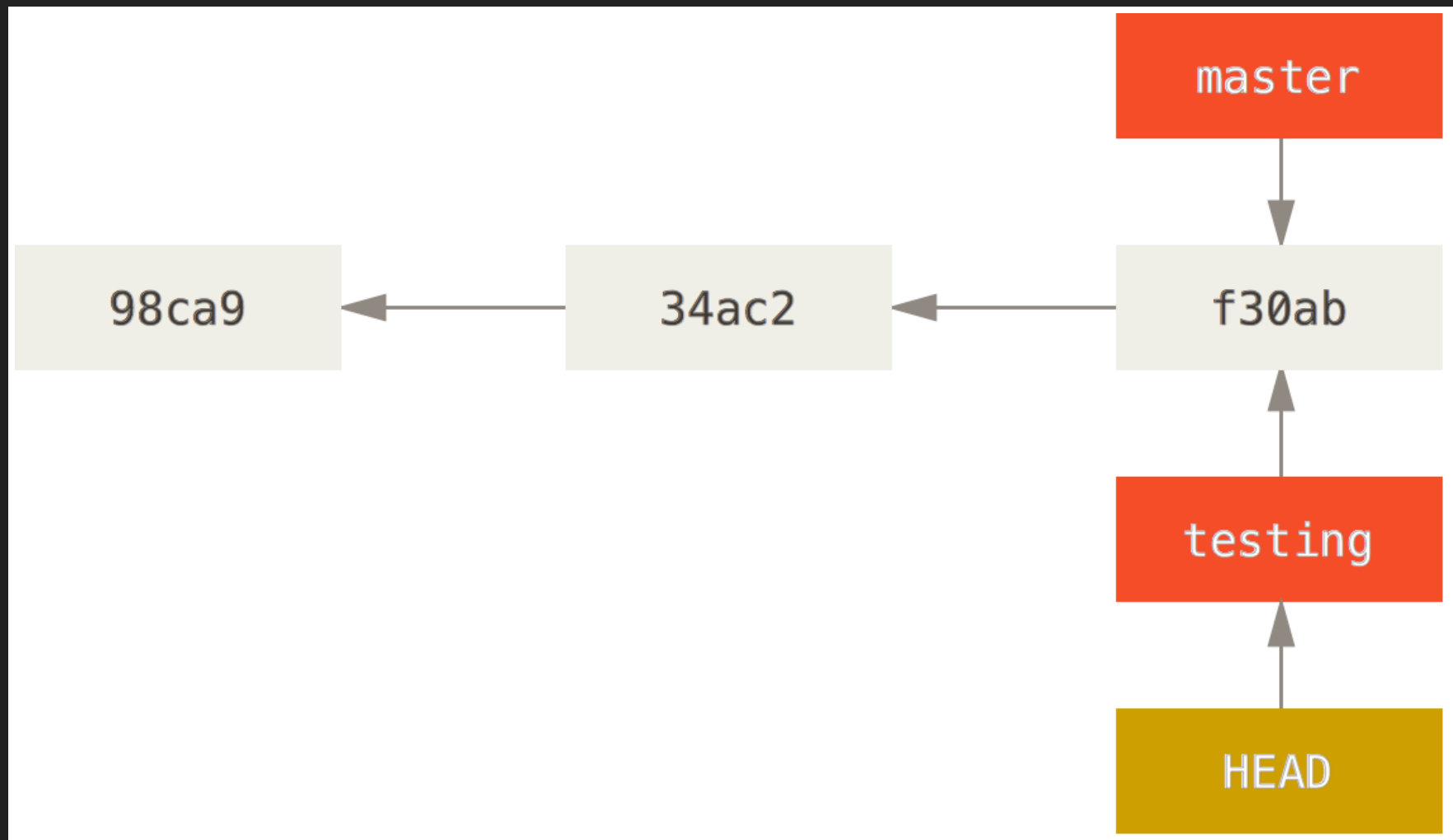




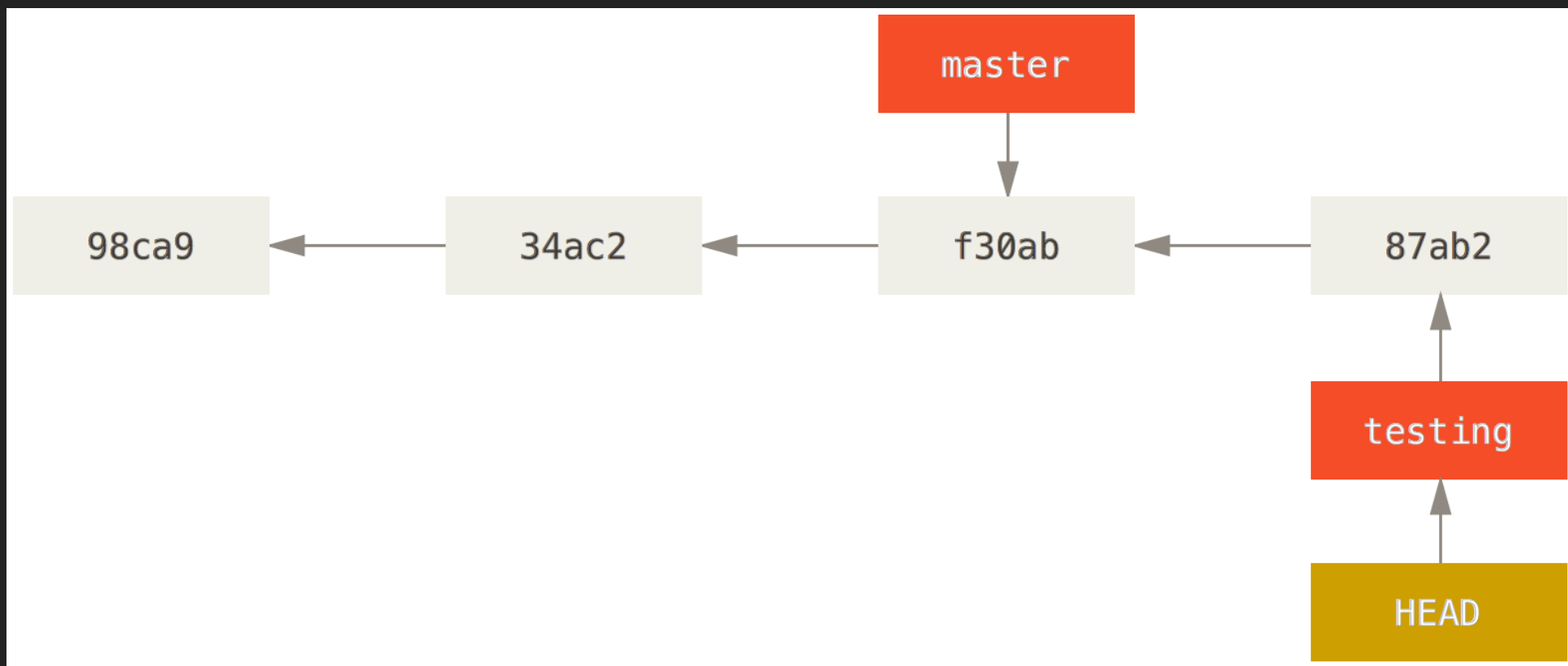








```
$ vim demo/test.php  
$ git commit -a -m "unit test setup"
```



# MERGES AND PATCHES



**MERGE**



```
$ git checkout master
$ git log --oneline --decorate --graph --all
* c34bc51 (testing) unit test setup
* db0d959 (HEAD -> master, origin/master, origin/HEAD)
```

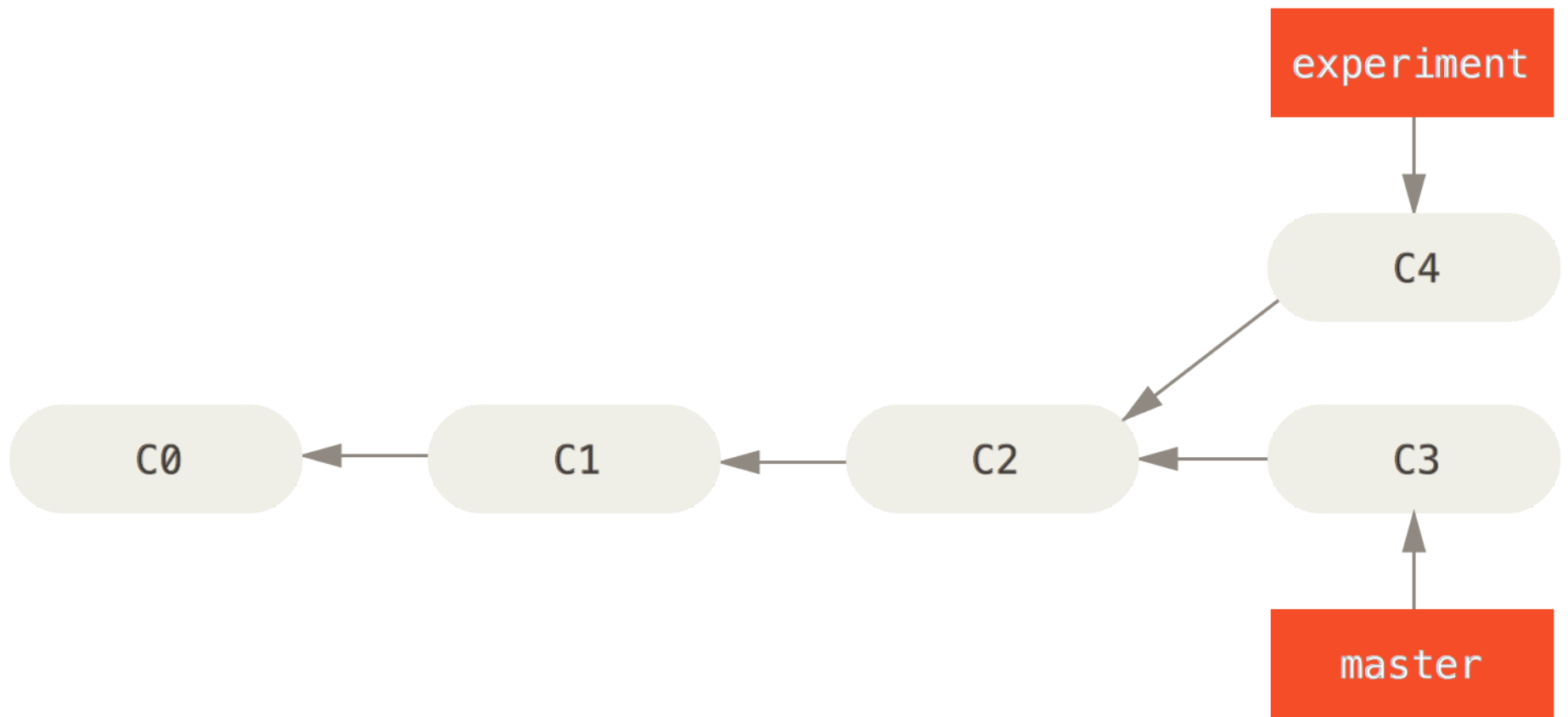
```
$ git merge testing
Updating 0fb644f..c34bc51
Fast-forward
 demo/test.php | 3 +++
 1 file changed, 3 insertions(+)
 create mode 100644 demo/test.php
```

## MERGE WITH CONFLICTS

- Theirs (*testing* branch)
- Mine (*master* branch)
- Merged (resulted file)

REBASE

FOR UNPUBLISHED WORK!  
ONE DOES NOT *REBASE* ON PUSHED  
COMMITTS!



---

```
$ git checkout experiment
```

```
$ git rebase master
```

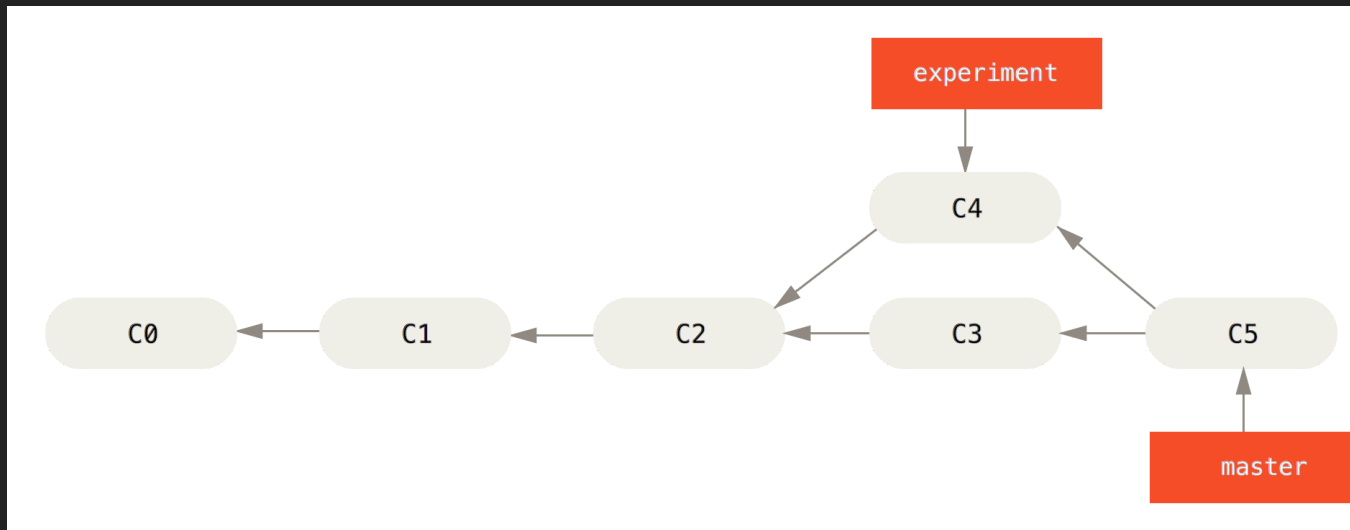
First, rewinding head to replay your work on top of it

Applying: added staged command

```
$ git checkout master
```

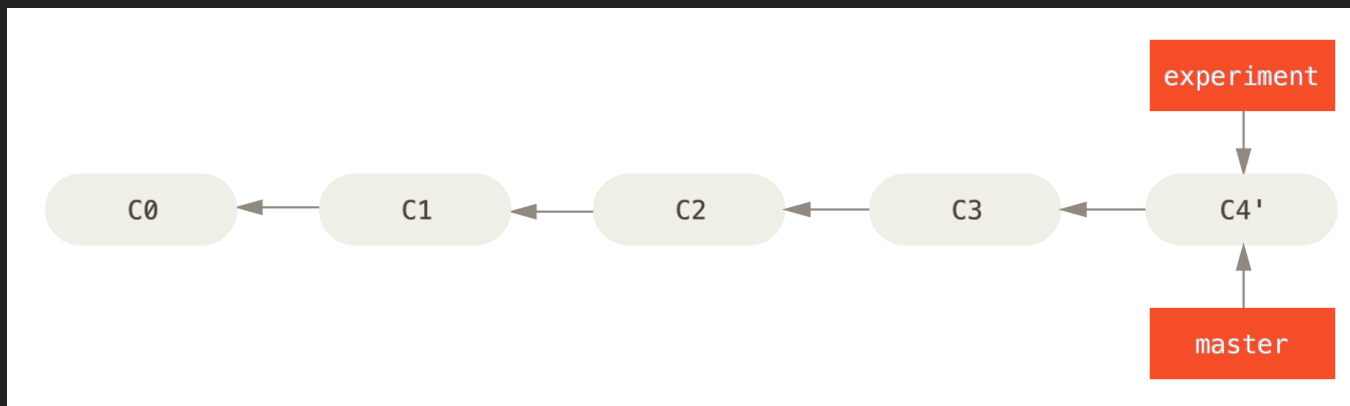
```
$ git merge experiment
```

# MERGE





# REBASE



# SOMETHING WENT WRONG

```
| * | | 2dc4f55 - Merge branch 'staging' into trend-grower-refactory (5 months ago) <Liviu Iosif>
| | \ \ \
| | * \ \ \ 0636269 - merge with staging (5 months ago) <Liviu Iosif>
| | | \ \ \
| | | * \ \ \ f7e5cea - Merge branch 'staging' into set-pin-modal-fields-agents (5 months ago) <Liviu Iosif>
| | | | \ \ \
| | | | * \ \ \ 7b6adac - Merge branch 'staging' into set-password-at-created-account (5 months ago) <Liviu Iosif>
| | | | | \ \ \
| | | | | * \ \ \ 54609e6 - merge with staging (5 months ago) <Liviu Iosif>
| | | | | | \ \ \
| | | | | | * \ \ \ 13df42f - Merge branch 'staging' into temptale-add-new-date-format (5 months ago) <Liviu Iosif>
| | | | | | | \ \ \
| | | | | | | * \ \ \ 9a7af22 - Merge branch 'staging' into cron-send-excel-imported-object-into-db (5 months ago) <Liviu Iosif>
| | | | | | | | \ \ \
| | | | | | | | * \ \ \ 00daeab - Merge branch 'staging' into cron-send-certificates-object-into-db (5 months ago) <Liviu Iosif>
| | | | | | | | | \ \ \
| | | | | | | | | * \ \ \ aa2bb81 - merge with staging (5 months ago) <Liviu Iosif>
| | | | | | | | | | \ \ \
| | | | | | | | | | * \ \ \ d2b8f77 - merge with staging (5 months ago) <Liviu Iosif>
| | | | | | | | | | | \ \ \
| | | | | | | | | | | * \ \ \ d6d5971 - Merge branch 'staging' into report-destination-field (5 months ago) <Liviu Iosif>
| | | | | | | | | | | | \ \ \
| | | | | | | | | | | | * \ \ \ 7964f3b - Merge branch 'staging' into price-band-info (5 months ago) <Liviu Iosif>
| | | | | | | | | | | | | \ \ \
| | | | | | | | | | | | | * \ \ \ 304d442 - modified: application/models/DbTable/PaymentStatus.php (5 months ago) <Liviu Iosif>
| | | | | | | | | | | | | | \ \ \
| | | | | | | | | | | | | | * \ \ \ fa65160 - set flag in ps_sellers for payment status cron for price updates (5 months ago) <Liviu Iosif>
| | | | | | | | | | | | | | | \ \ \
| | | | | | | | | | | | | | | * \ \ \ e89903b - price updates per pallets modified: application/models/DbTable/PaymentStatus.php (5 months ago) <Liviu Iosif>
| | | | | | | | | | | | | | | | \ \ \
| | | | | | | | | | | | | | | | * \ \ \ 2354b4f - ... (5 months ago) <Razvan Zofota>
| | | | | | | | | | | | | | | | | \ \ \
| | | | | | | | | | | | | | | | | * \ \ \ 086f1a1 - modified: application/library/Code932/Traits/Export.php (5 months ago) <Liviu Iosif>
| | | | | | | | | | | | | | | | | | \ \ \
| | | | | | | | | | | | | | | | | | * \ \ \ b3b8478 - modified: application/js/code932/chart-data-img.php (5 months ago) <Liviu Iosif>
```

**CHERRY-PICK**

---

```
$ git cherry-pick <commit hash>  
$ git cherry-pick 086f1a1
```

---

# PASSWORD CACHING

```
# only for 15 min, but can be changed
$ git config [--global] credential.helper cache [--timeout <seconds>

# permanent, but stored on disk
$ git config [--global] credential.helper 'store --file <credentials-file>'

$ cat <credentials-file>
https://iionut:<password>@github.com
```

**Mac:** *osxkeychain* mode (already installed)

**Windows:** *wincred* helper (can be installed)

# BEST PRACTICES

1. Commit often
2. Commit related changes
3. Write clear and expressive commit messages
4. Don't use only one branch
5. Don't panic
6. Don't change published history
7. Don't commit big, unnecessary files (media, archives, .vagrant/, logs, vendor/, bower\_components/, etc.)
8. Commit configuration/declaration files (Makefile, composer.json, etc.)
9. Don't commit half-done work (stash it)
10. Version control is not a backup system
11. Git doesn't do anything by itself

# LEARNING MATERIAL

<https://git-scm.com/book/en/v2>

<https://www.atlassian.com/git/tutorials/>



**GOOD DAY!**