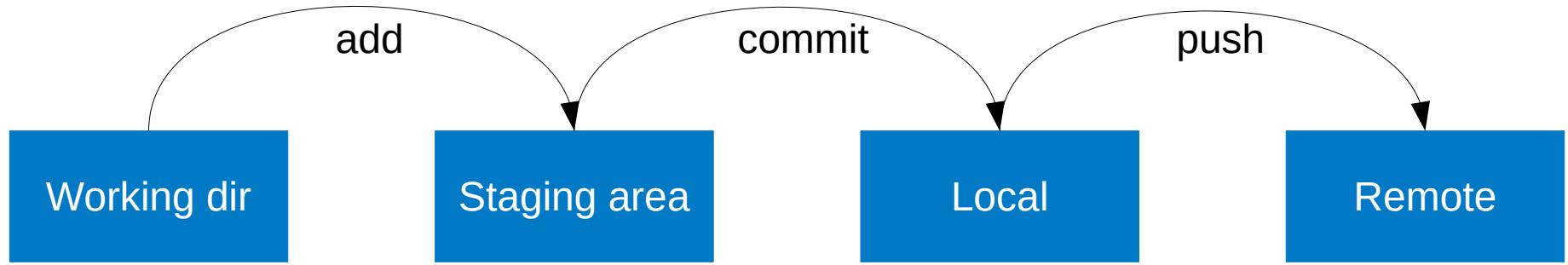


Beyond git add/commit/push



Based on original version made by
Alexis López @aa_lopez

About us



@tuxtor



github.com/tuxtor



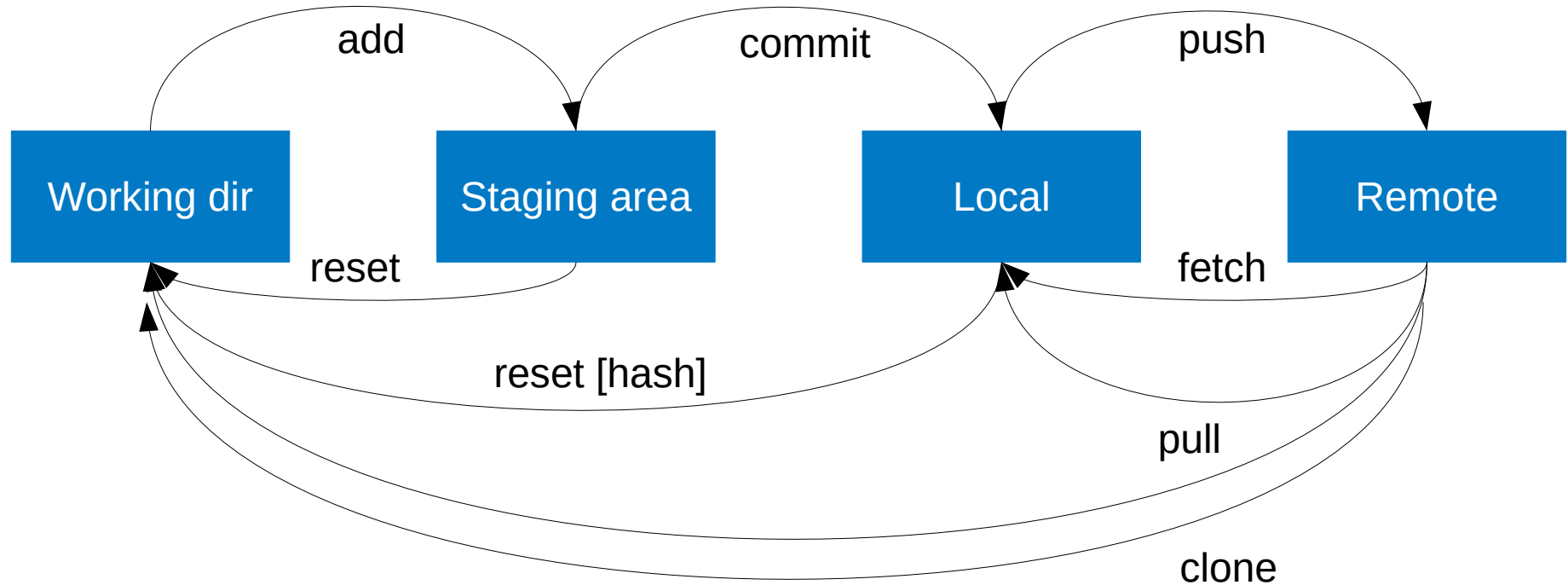
@edivargas



jorgevargas.mx



Review traditional commands



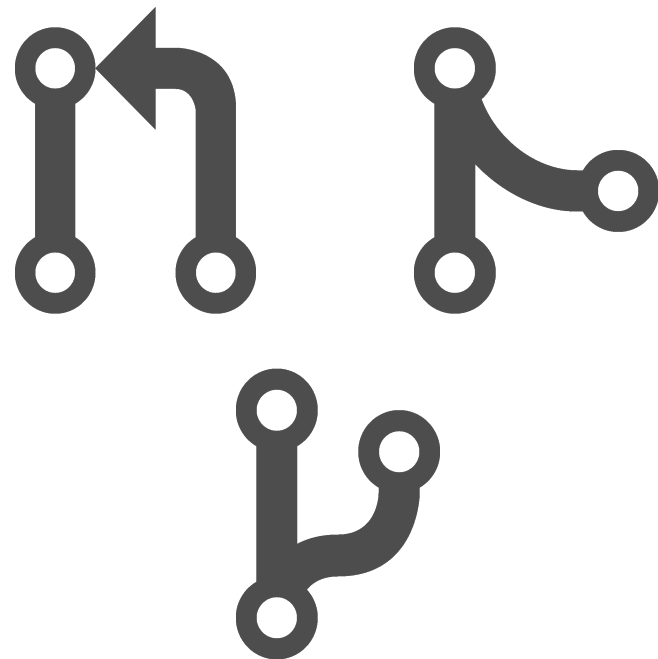
Git System

- Users
- Repo management
 - Branches
 - Tags
 - History
 - Create/Edit files
- Merge Requests
 - Notifications
 - Collaboration
 - Interaction

Git Workflows

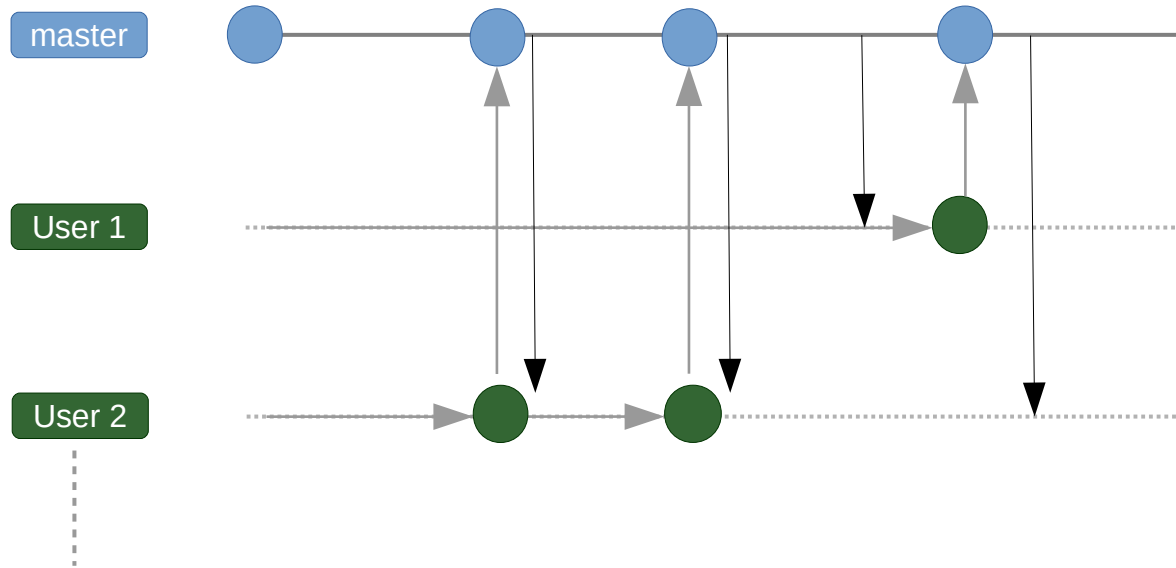
Definition

- Guides on how to work with a Git repo.
- It's a recommendation
- Team members should know them and follow them if apply
- There are different workflows for different projects



Git Workflows

Centralized

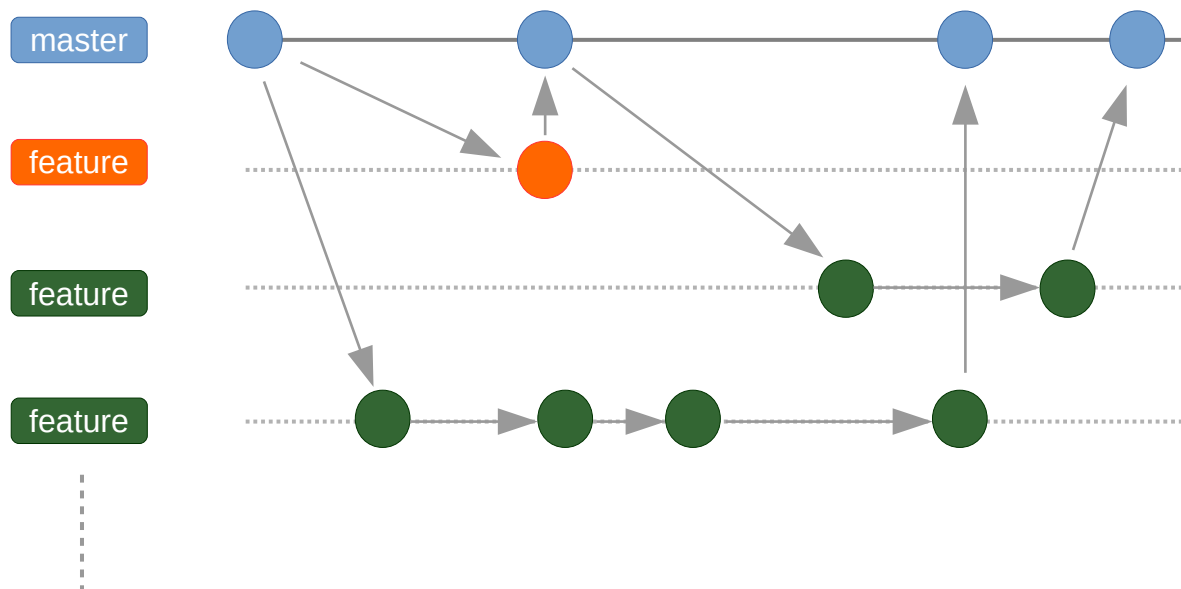


New git users

Migrating from
SVN systems

Change trunk label
to master and
begins to learn

Feature Branch



Strong branch use

Base for other workflows

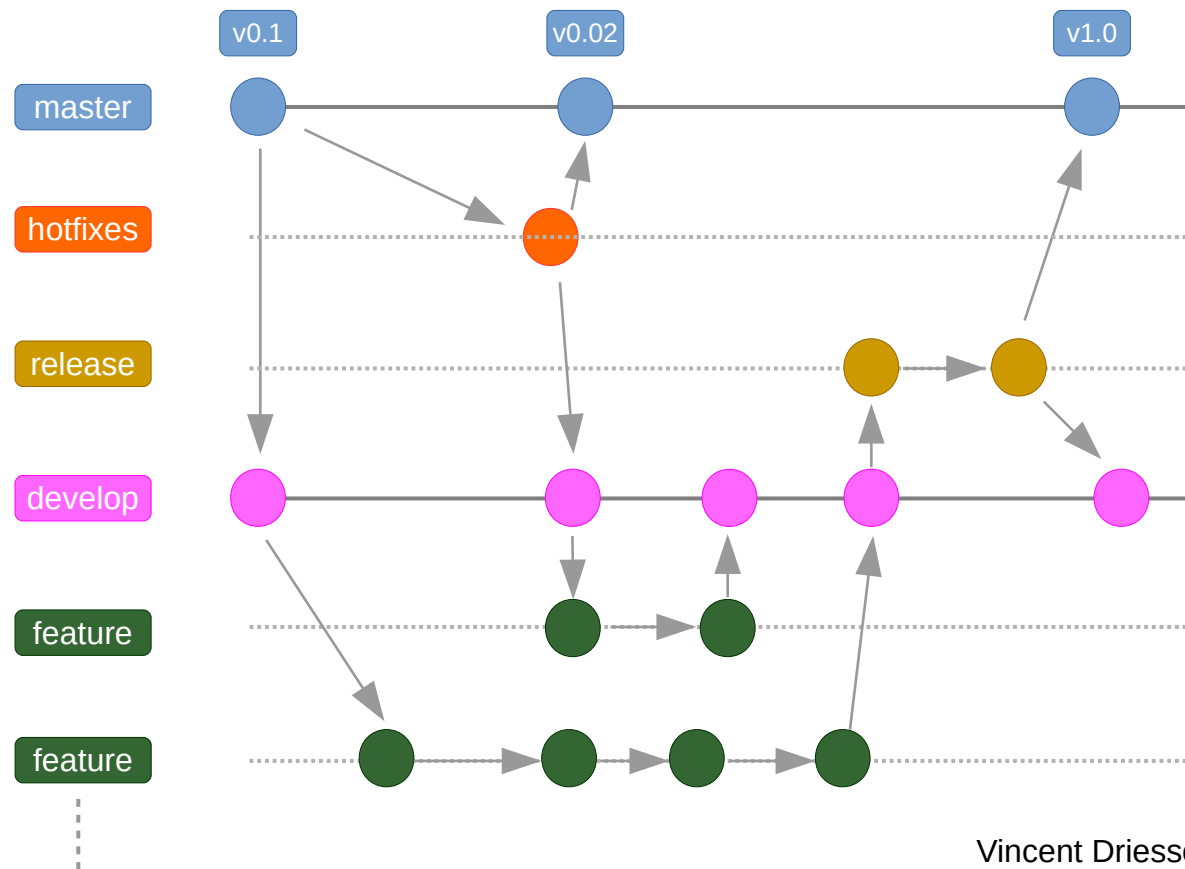
Encourages collaboration

Merge Requests

Code Review

Git Workflows

Gitflow



Defines branches
role and their
interaction

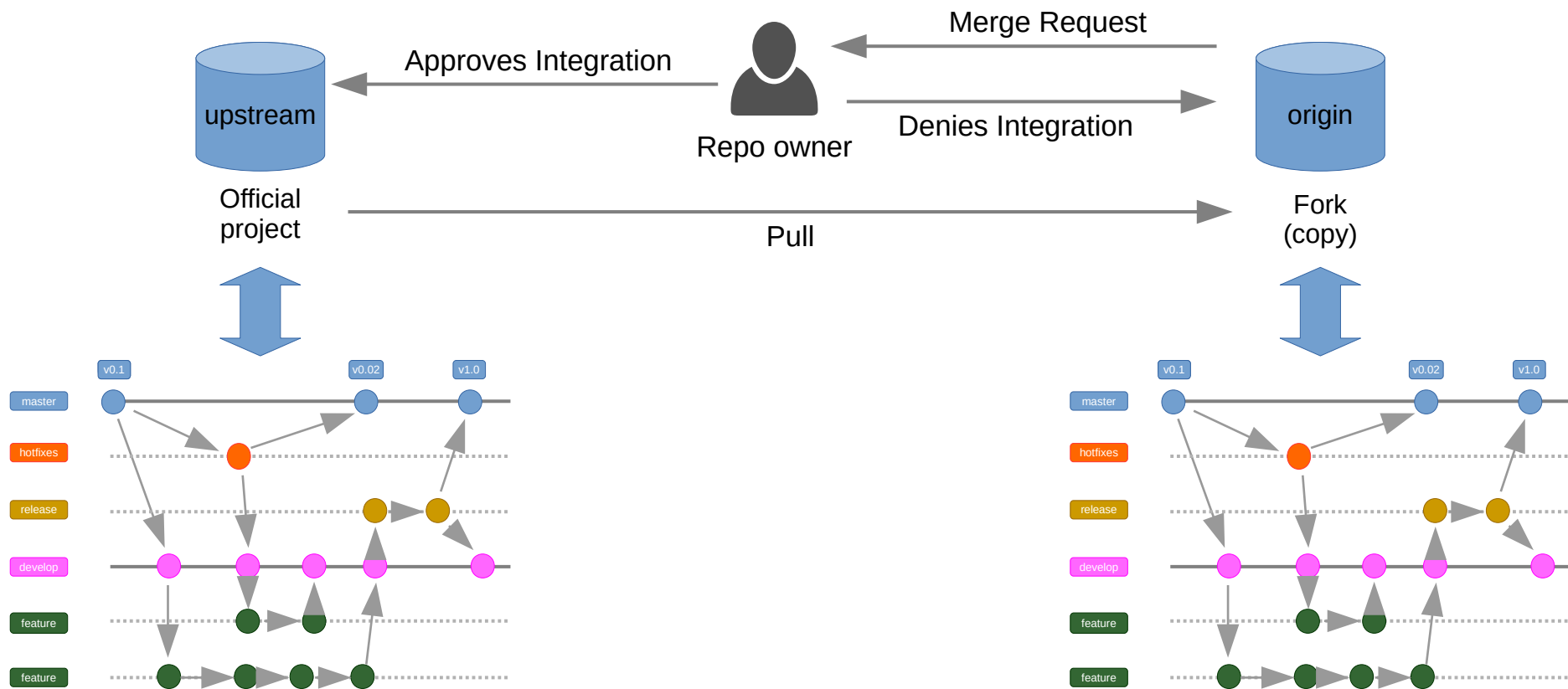
Useful for projects
that follows a
release model

Extension: git-flow

Vincent Driessen - <http://nvie.com/posts/a-successful-git-branching-model/>

Git Workflows

Forking



Git Workflows

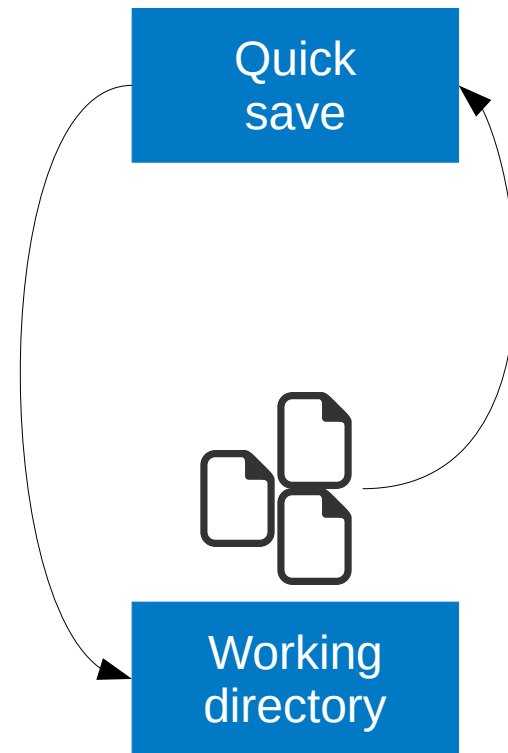
Recommendations

- There is no one workflow to rule them all
- Business model influences the selection of a workflow
- Code in master branch should be complete and functional at all times
- Create short-lived branches
- Use meaningful names for branches
- Use meaningful commit messages
- Each commit should be complete and coherent

Git Advanced Commands

git stash

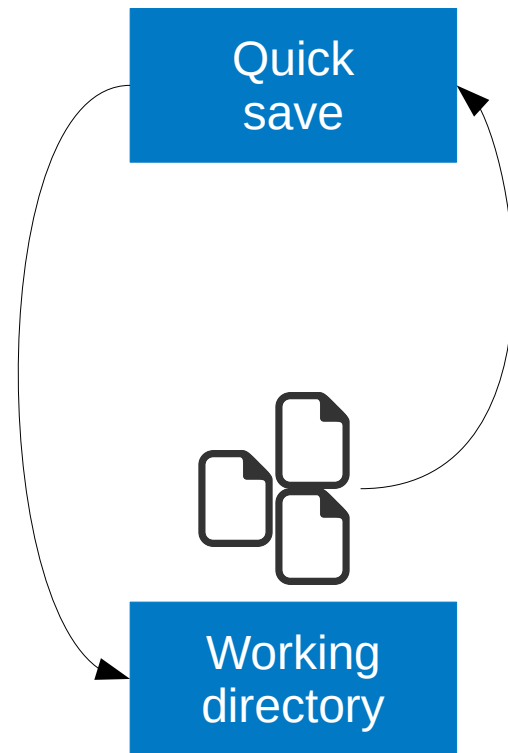
- Quick save for the working directory.
- Allows you to work in something else, and then come back and continue.
- No commits.
- You can re-apply changes to your working directory later on.
- Only local.



Git Advanced Commands

git stash – Recommendations

- **git save** is deprecated in recent versions, use **git push** instead.
- Use meaningful messages for your stash entries.
- Use short-lived stash entries.
- ¿Conflicts? Use:
`git stash branch new-branch stash@{n}`



Git Advanced Commands

git stash – Quick save

Quick save with a generic message (WIP + branch + hash).
Warning! This does not saves changes on ignored or untracked files.

git stash

Quick save with the specified message.
Warning! This does not saves changes on ignored or untracked files.

git stash save "message"

Quick save with the specified message.
Includes untracked files, but not ignored files.

git stash save -u "message"

Quick save with the specified message.
Includes untracked and ignored files.

git stash save -a "message"

Git Advanced Commands

git stash – List/Show stash entries

Lists stash entries.

git stash list

Shows a summary of the changes recorded in the most recent stash entry.

git stash show

Shows a summary of the changes of the nth stash entry.

git stash show stash@{n}

Shows detailed information about the changes in the most recent stash entry.

git stash show -p

Shows detailed information about the changes of the nth stash entry.

git stash show -p stash@{n}

Git Advanced Commands

git stash – Apply/Drop stash entries

Apply and drop the most recent stash entry.

You can also apply and drop the nth stash entry.

git stash pop [stash@{n}]

Apply and drop the most recent stash entry, does not drop it.

You can also apply the nth stash entry.

git stash apply [stash@{n}]

Drops the most recent stash entry.

You can also drop the nth stash entry.

git stash drop [stash@{n}]

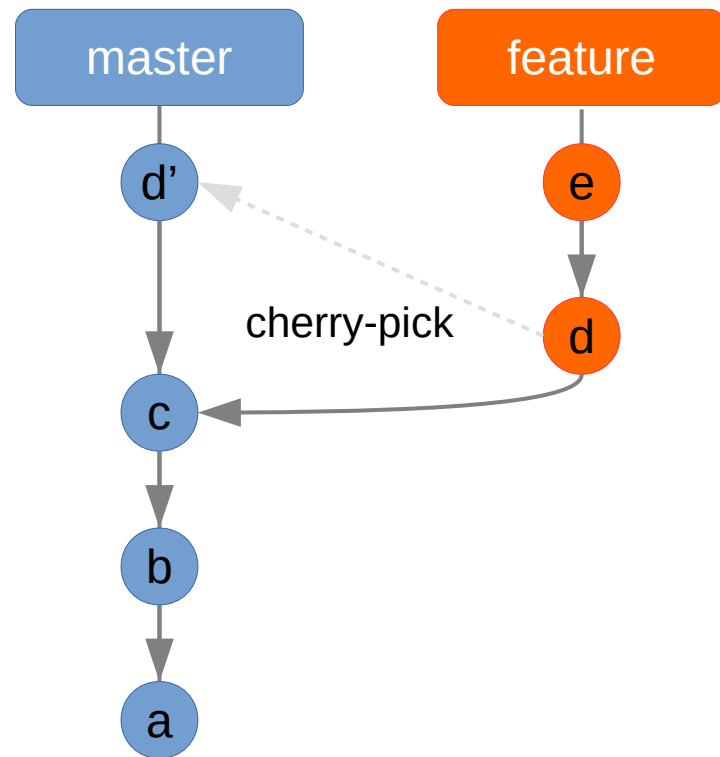
Drops all stash entries.

git stash clear

Git Advanced Commands

git cherry-pick

- Selects and copies a commit into the current branch.
- In case of conflicts, solve them and then you can continue/abort.
- New commits with new IDs for each copied commit.

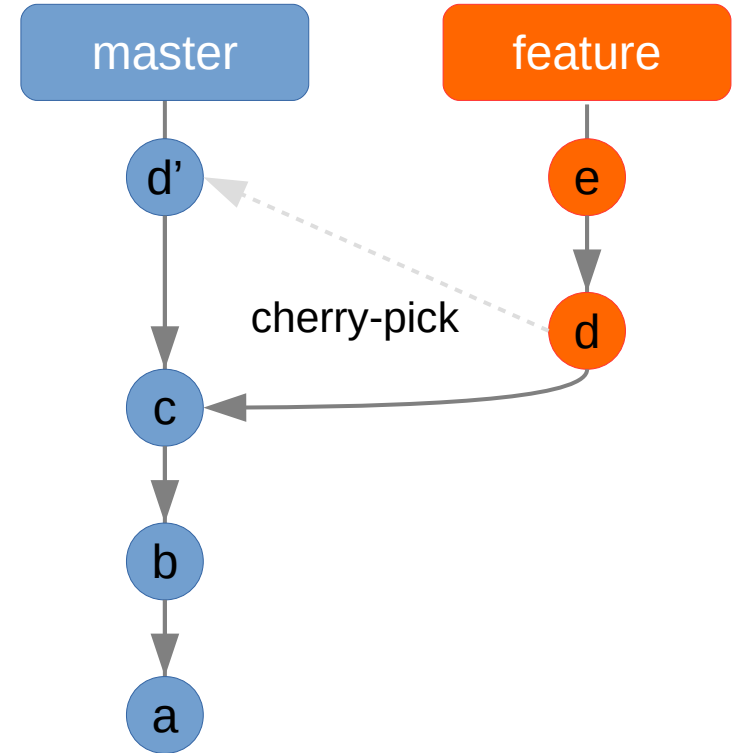


Git Advanced Commands

git cherry-pick

1. Checkouts target branch, master in this case
`git checkout master`

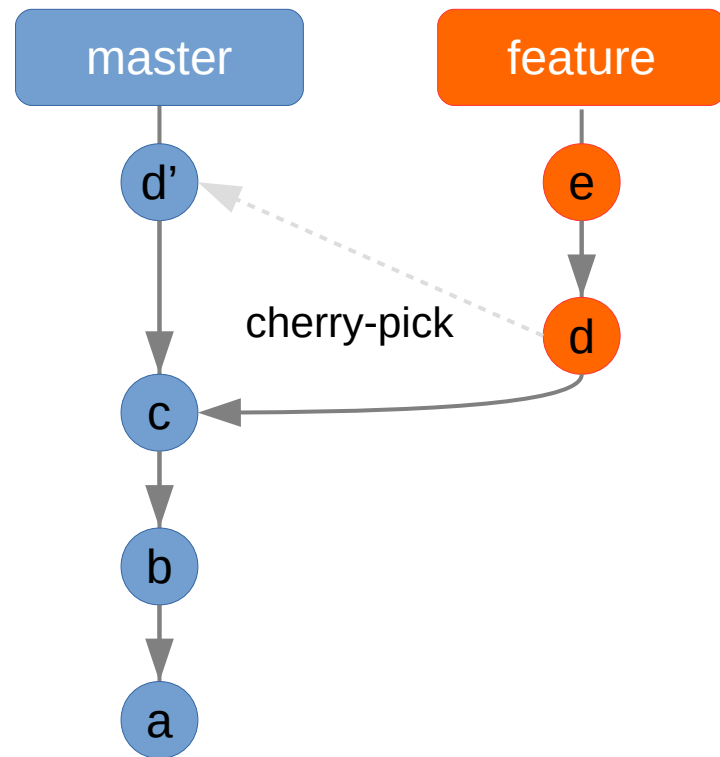
2. Copies commit from feature branch to master branch
`git cherry-pick -x commitId`



Git Advanced Commands

git cherry-pick - Recommendations

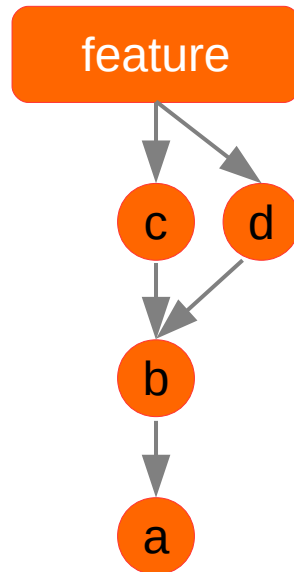
- Watch out! Duplicated commits are created.
- Use **-x** to auto. add the following to the commit message:
 - "(cherry picked from commit ...)"
- If more than one consecutive commit is needed, better use **merge** or **rebase**.



Git Advanced Commands

git commit --amend

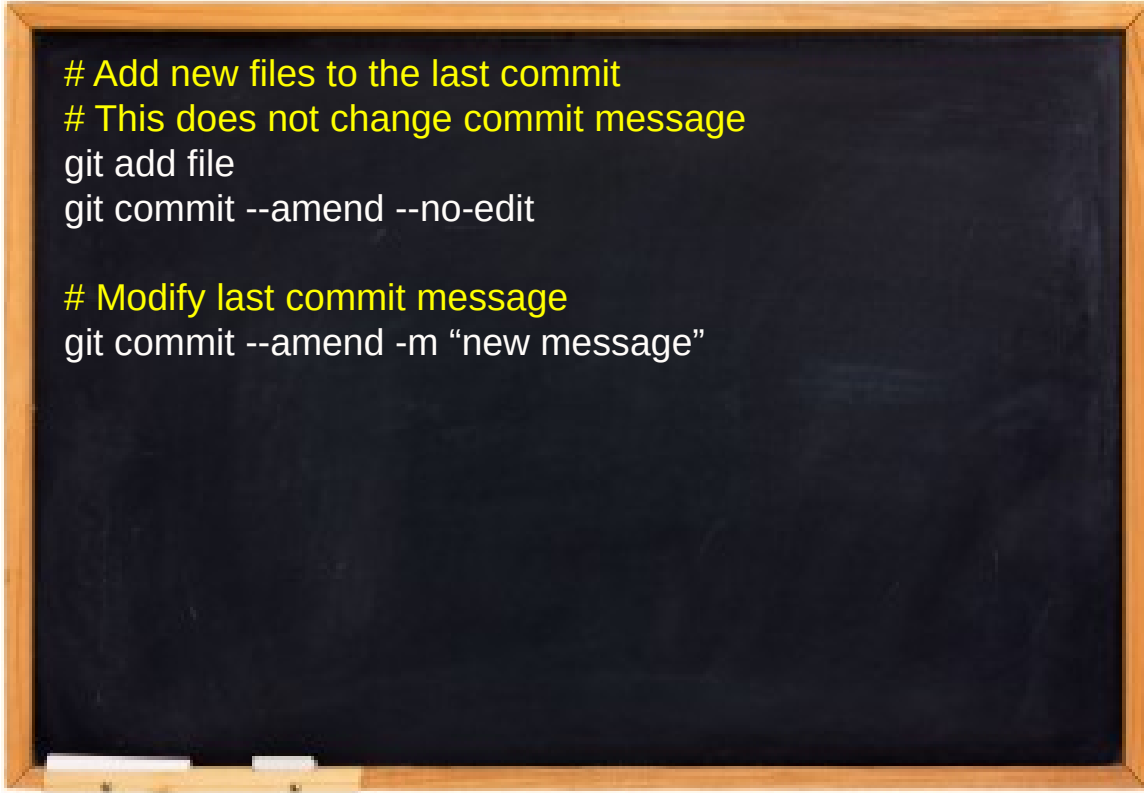
- Lets you “modify” the last commit:
 - Add files from the staging area.
 - Change commit message.
- By default, it creates a new commit using same author/timestamp.



git commit --amend

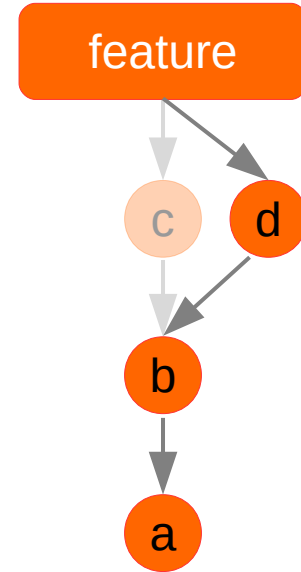
Git Advanced Commands

git commit --amend



```
# Add new files to the last commit
# This does not change commit message
git add file
git commit --amend --no-edit

# Modify last commit message
git commit --amend -m "new message"
```

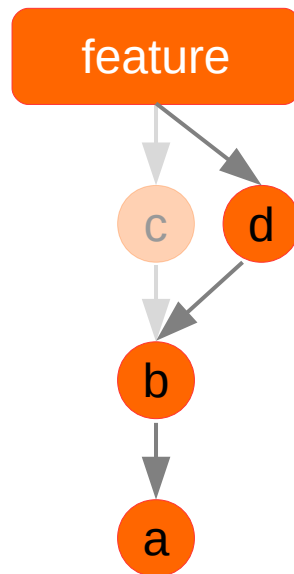


git commit --amend

Git Advanced Commands

git commit --amend - Recommendations

- Gold Rule: Do not modify commits that have already been shared.
 - **git push -f origin branch-name**
make sure you understand the risks...



git commit --amend

Git Advanced Commands

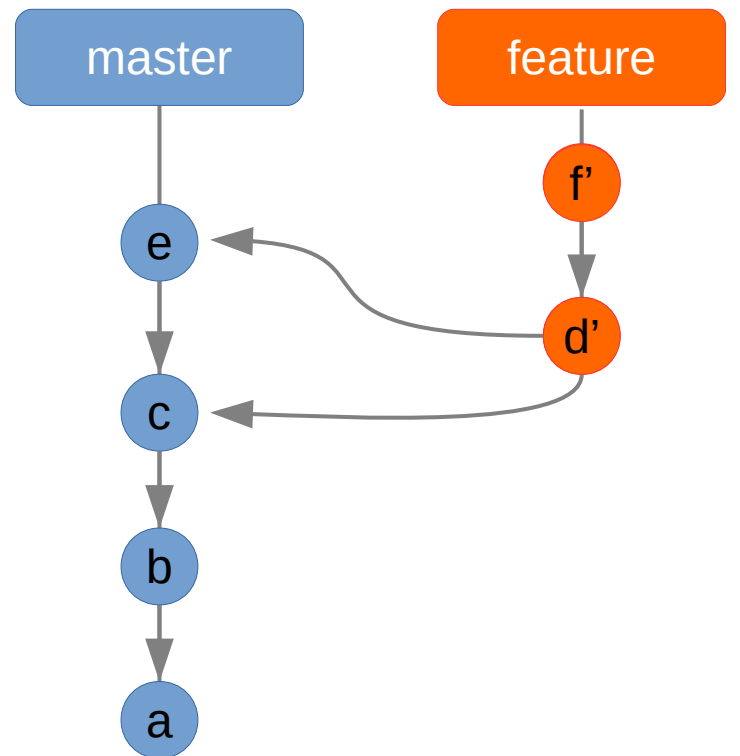
git rebase



Git Advanced Commands

git rebase

- Allows you to rewrite history.
- It's also used to integrate changes from one branch into another:
 - Linear history.
 - As if the branch was created from the latest changes.
- Two modes:
 - Manual
 - Interactive:
 - Amend/combine/drop commits...



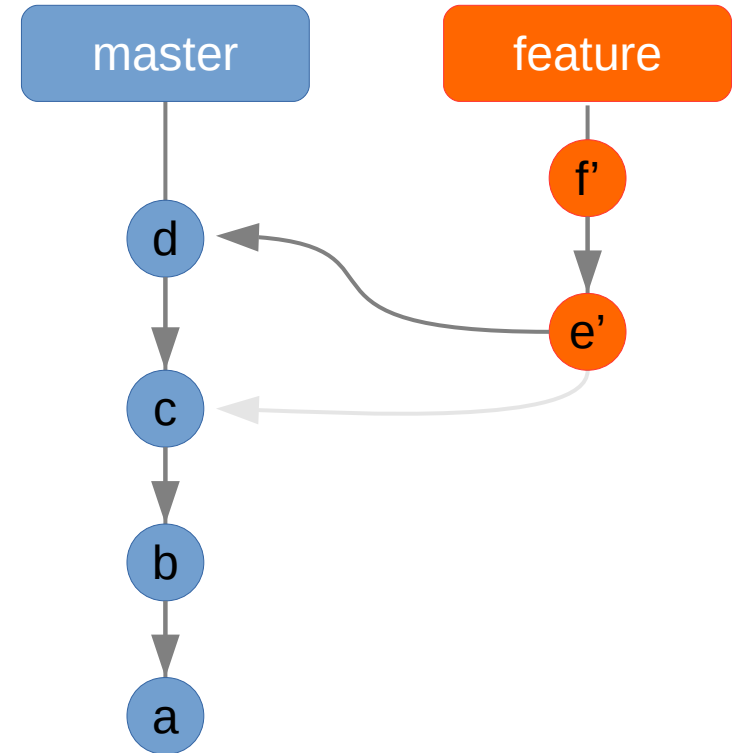
git rebase master feature

Git Advanced Commands

git rebase

Modifies second branch base so it is as if it was
created from the tip of master (Linear History)
git rebase master feature

Modifies current branch history from commit n
git checkout feature
git log
git rebase -i HEAD~n

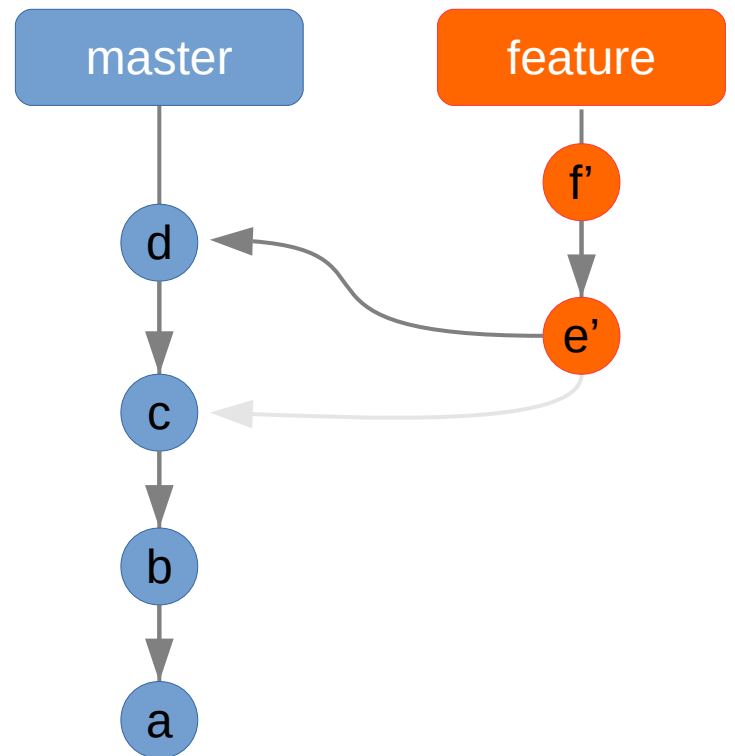


git rebase master feature

Git Advanced Commands

git rebase - Recommendations

- Gold Rule: Do not modify commits that have already been shared.
 - **git push -f origin branch-name** make sure you understand the risks...



git rebase master feature

Links

More content...

- Git official book
<https://git-scm.com/book>
- Git tutorials
<https://www.atlassian.com/git/tutorials/what-is-version-control>
- Developer Cloud Service
https://cloud.oracle.com/developer_service
- Icons
<https://www.iconfinder.com>

Thank you!!!



@tuxtor



github.com/tuxtor



@edivargas



jorgevargas.mx

