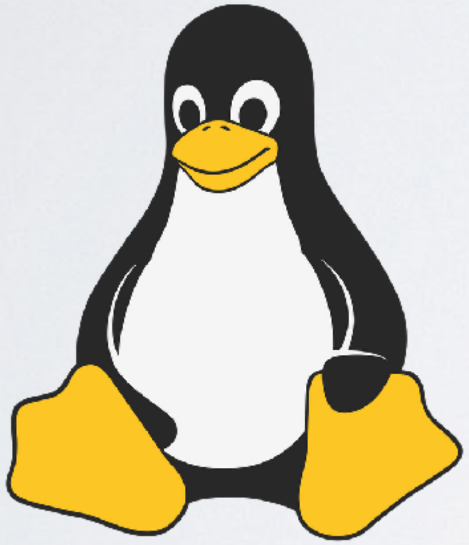
Develop, Collaborate, and Manage

USING GIT

A Brief Introduction to Git

Linus Torvads



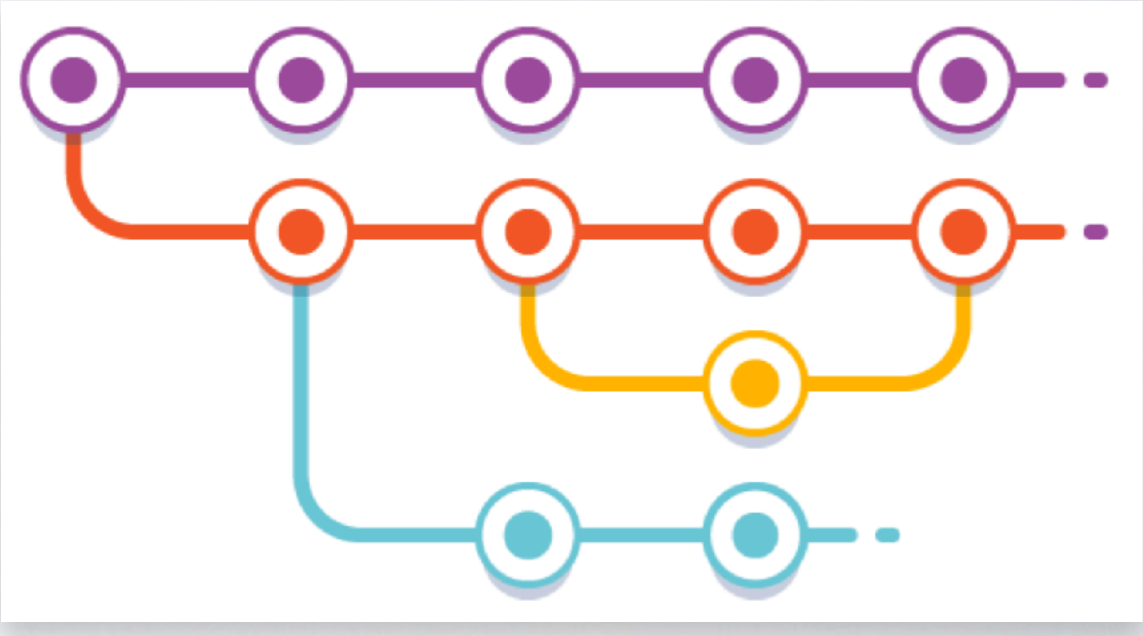
Key Ideas of Git

> *"In many ways you can just see git as a filesystem..."*
>
> — *Linus Torvalds*

- Journaling: commits, branching, history, etc.
- Caching: stage, local and remote, etc.

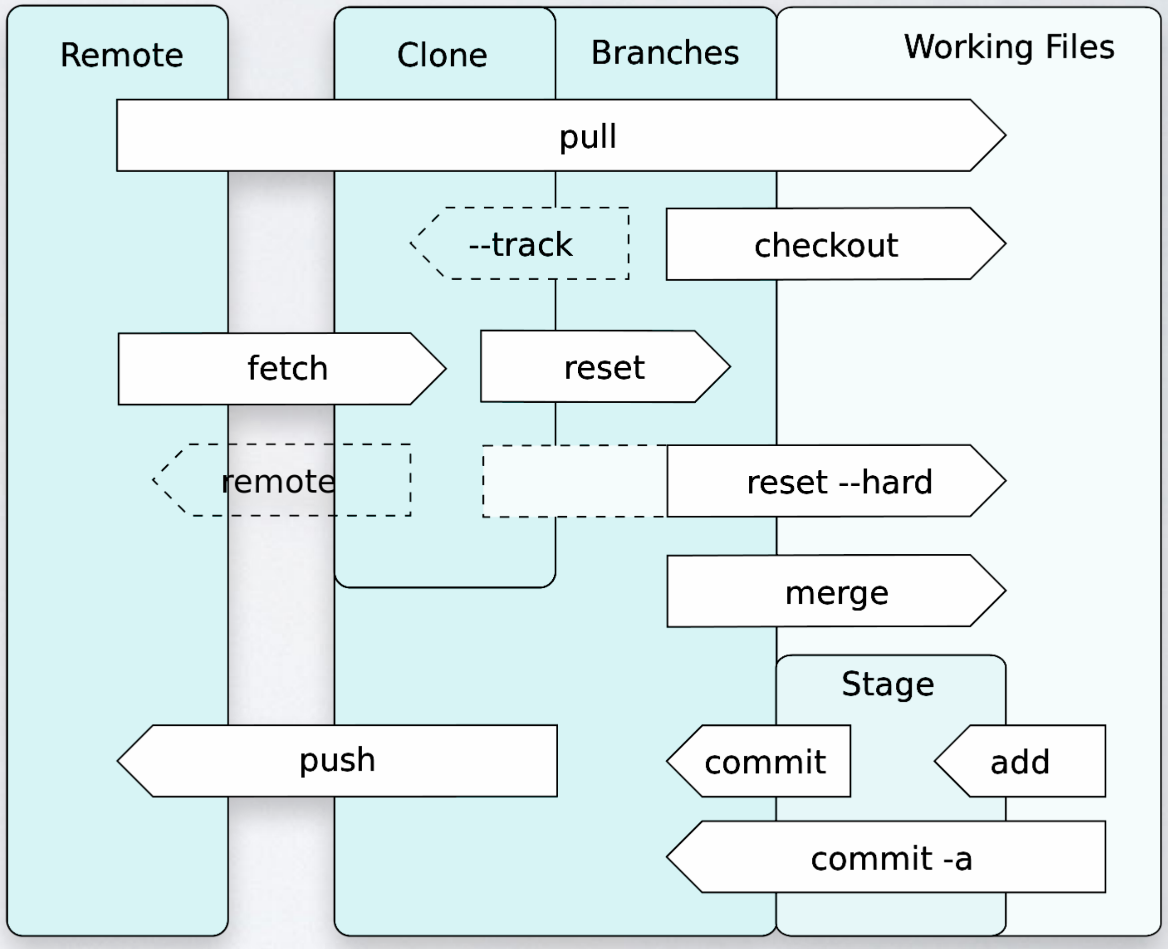
Journaling

- Version control: **commits**
- Nonlinear history: branching and merging
- Refs and **HEAD**

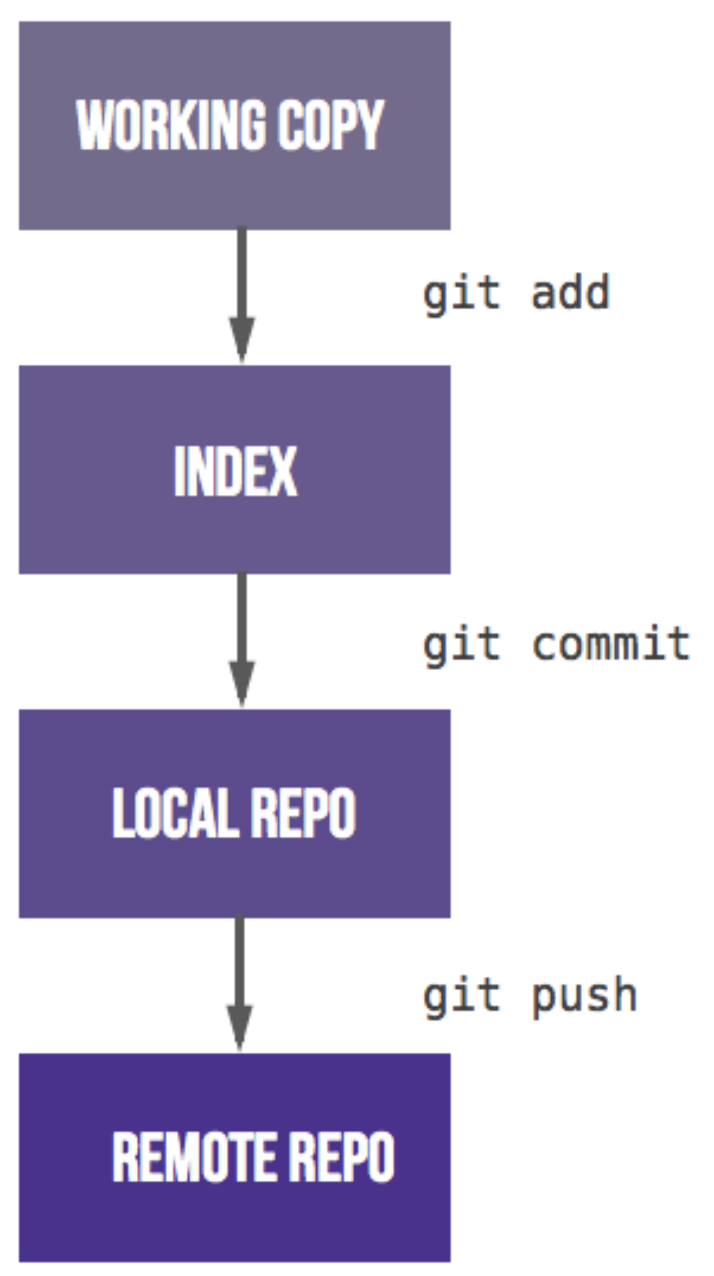


Caching

- Working files
- Stage (index, cache)
- Local Repo
- Remote Repo



Personal Use: Code Backup



WORKING COPY

git add

INDEX

git commit

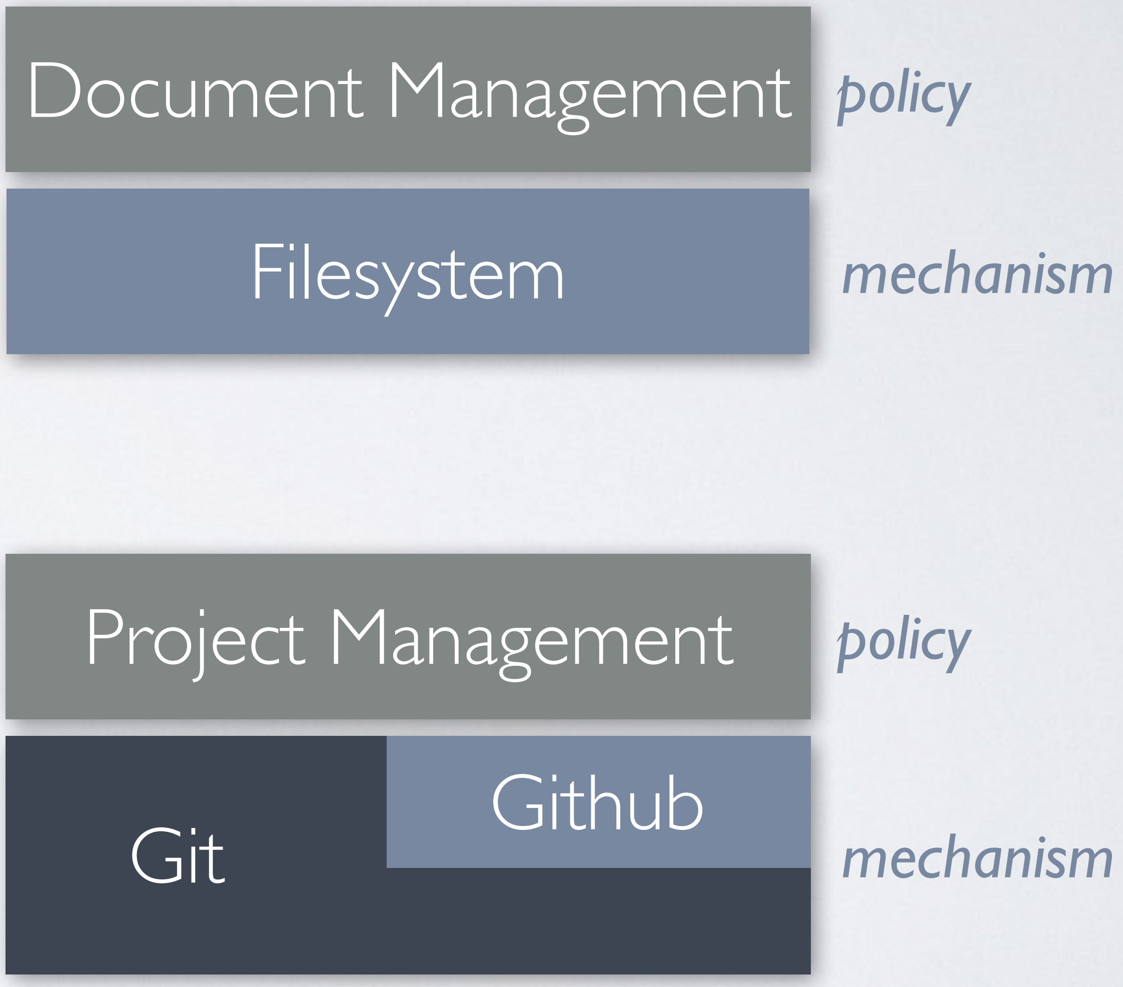
LOCAL REPO

git push

REMOTE REPO

Mechanism vs Policy

- Git provides **mechanism** for version control
- **Policy** for project management: **workflow**



Document Management *policy*

Filesystem *mechanism*

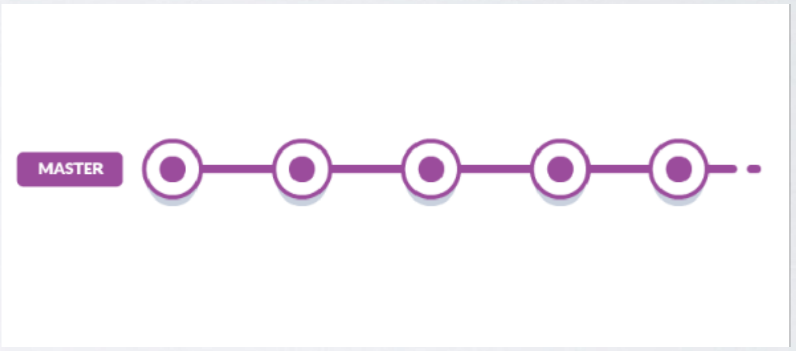
Project Management *policy*

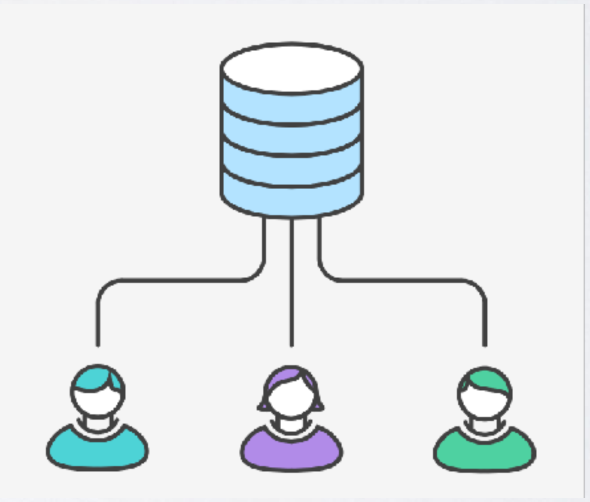
Git Github *mechanism*

Git Workflow

Linear Workflows

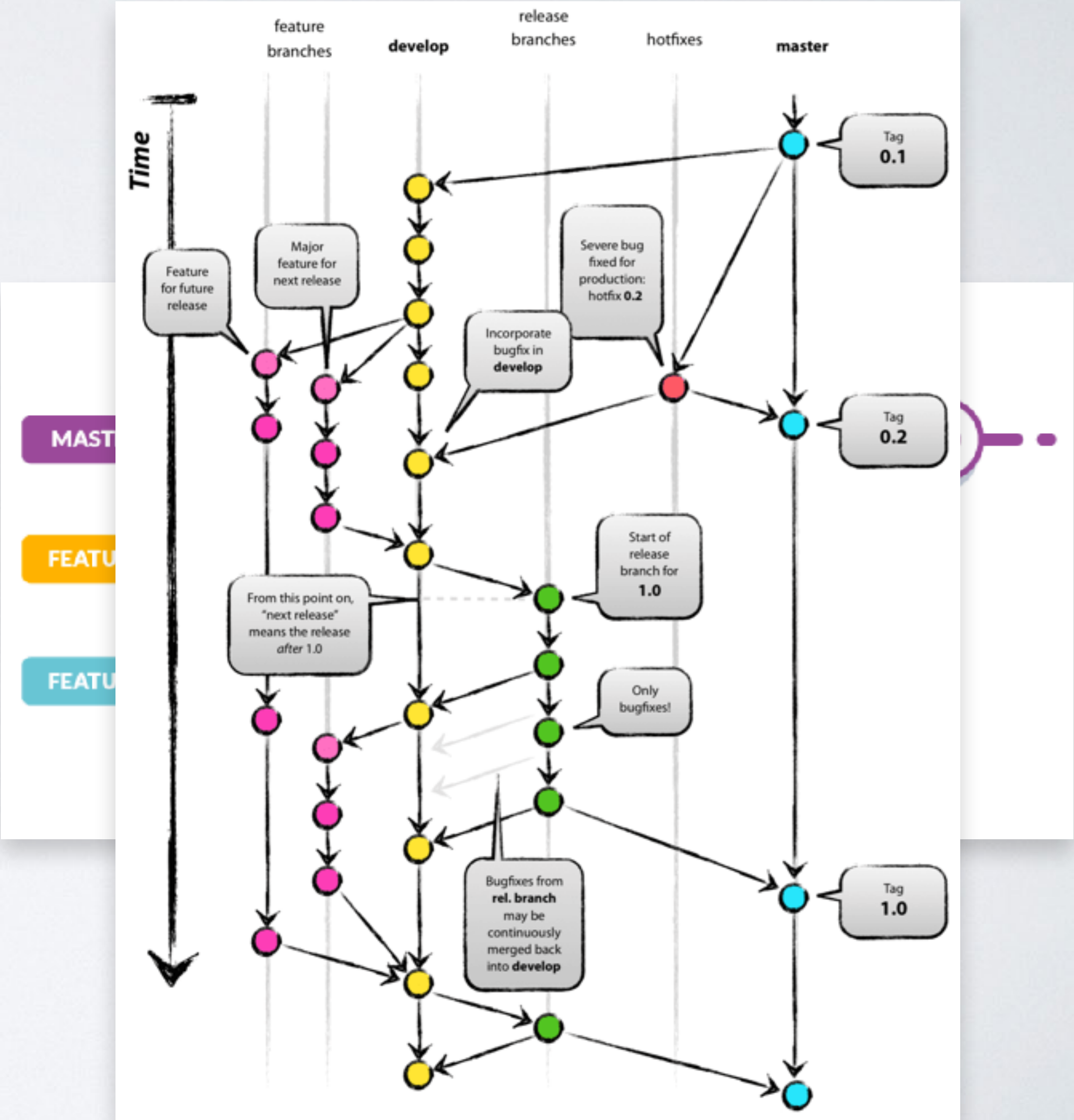
- **Linear** development
- Centralized management





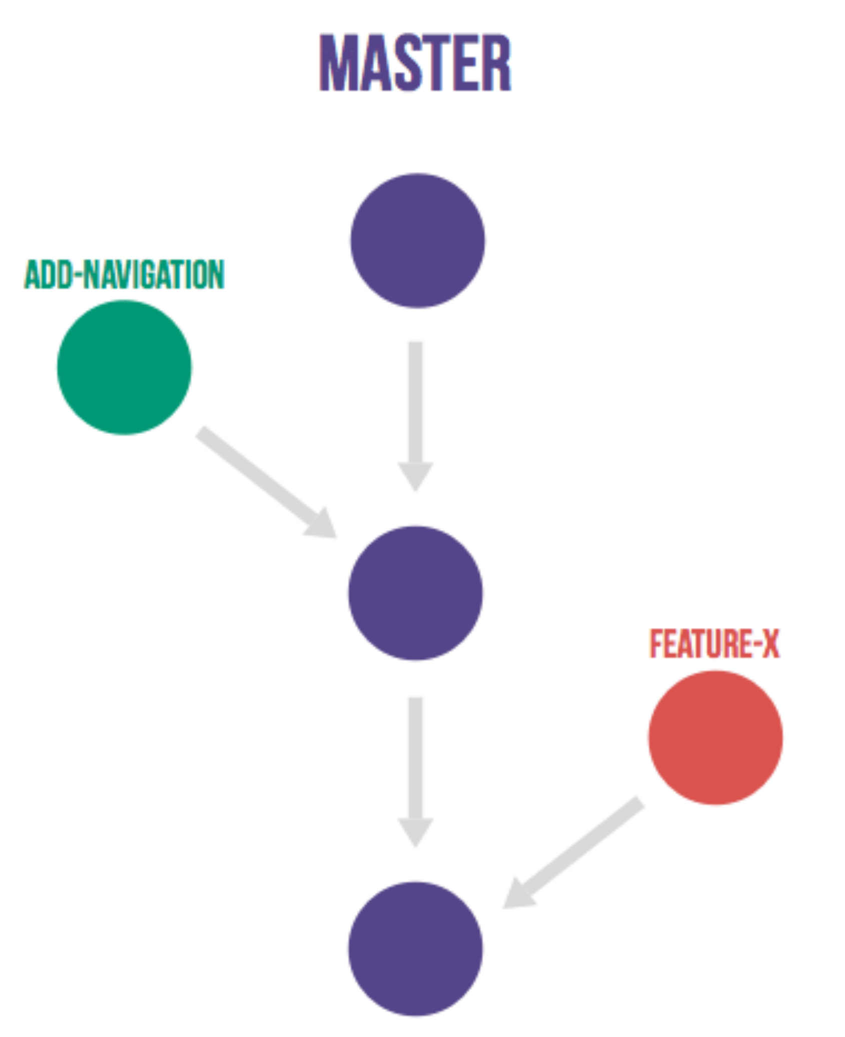
Feature Based Workflows

- **Feature**: the concept
- Feature as **branch**
- **Nonlinear** history


MAST
FEATU
FEATU
feature branches
develop
release branches
hotfixes
master
Time
Tag 0.1
Severe bug fixed for production: hotfix 0.2
Major feature for next release
Feature for future release
Incorporate bugfix in develop
Tag 0.2
Start of release branch for 1.0
From this point on, "next release" means the release after 1.0
Only bugfixes!
Bugfixes from rel. branch may be continuously merged back into develop
Tag 1.0

Github and GitLab Workflow

- Management: **merge request**
- **Sub-linear** history
- Continuous Integrity (**CI**)
- Collaboration with global community: **forking**



BDAA Workflow

BDAA Workflow

- For **closer** groups
- Well-defined and simplified routines
- Flexible project management

Main Ideas

- Developing: feature branch and working branch
- Collaboration: simplified **Sync-Test-Merge** routine
- Management: testing and reviewing, history maintenance

Developing

- Start from a working branch:

```
$ git checkout wip/working-branch

$ git pull
```

- Create a feature branch:

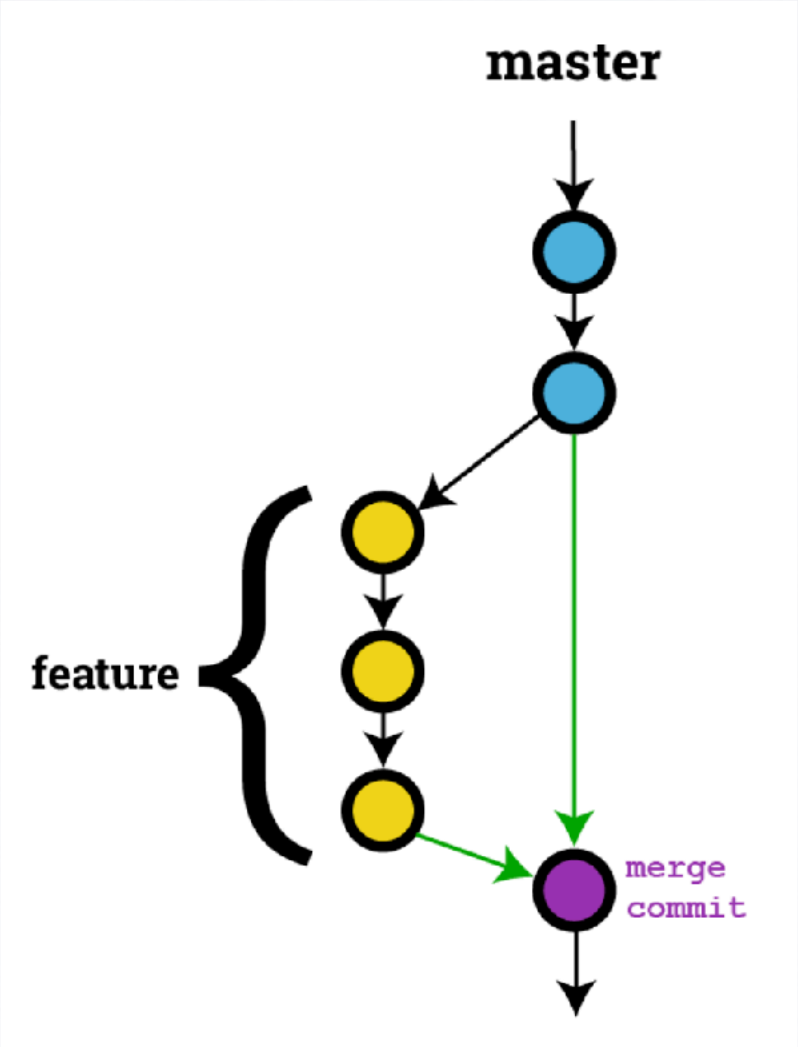
```
$ git checkout -b feature-1
```

- Code and commit:

```
$ git add .

$ git commit -m 'some meaningful message'
```

Developing



Collaboration

The **Sync-Test-Merge** routine:

- Sync: make sure your changes are on top of others

```
$ git fetch

$ git rebase origin/wip/working-branch
```

- Test: **always** test before publish your work

```
$ pytest  # for example
```

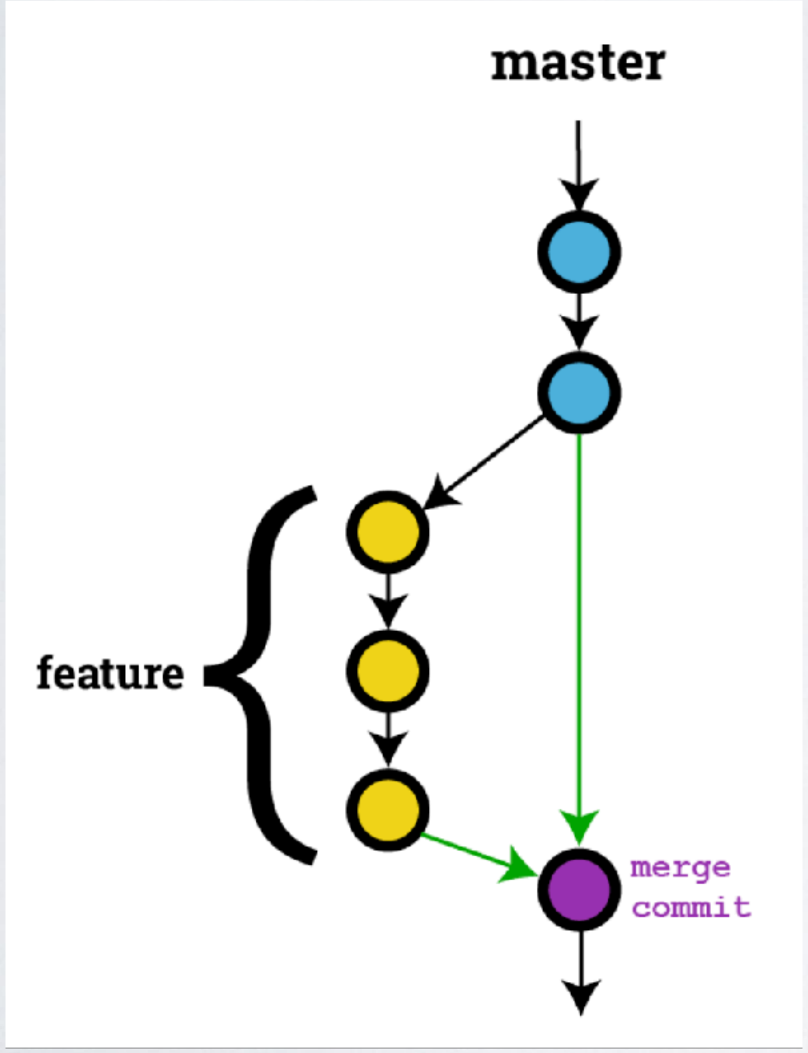
- Merge: push your work to remote

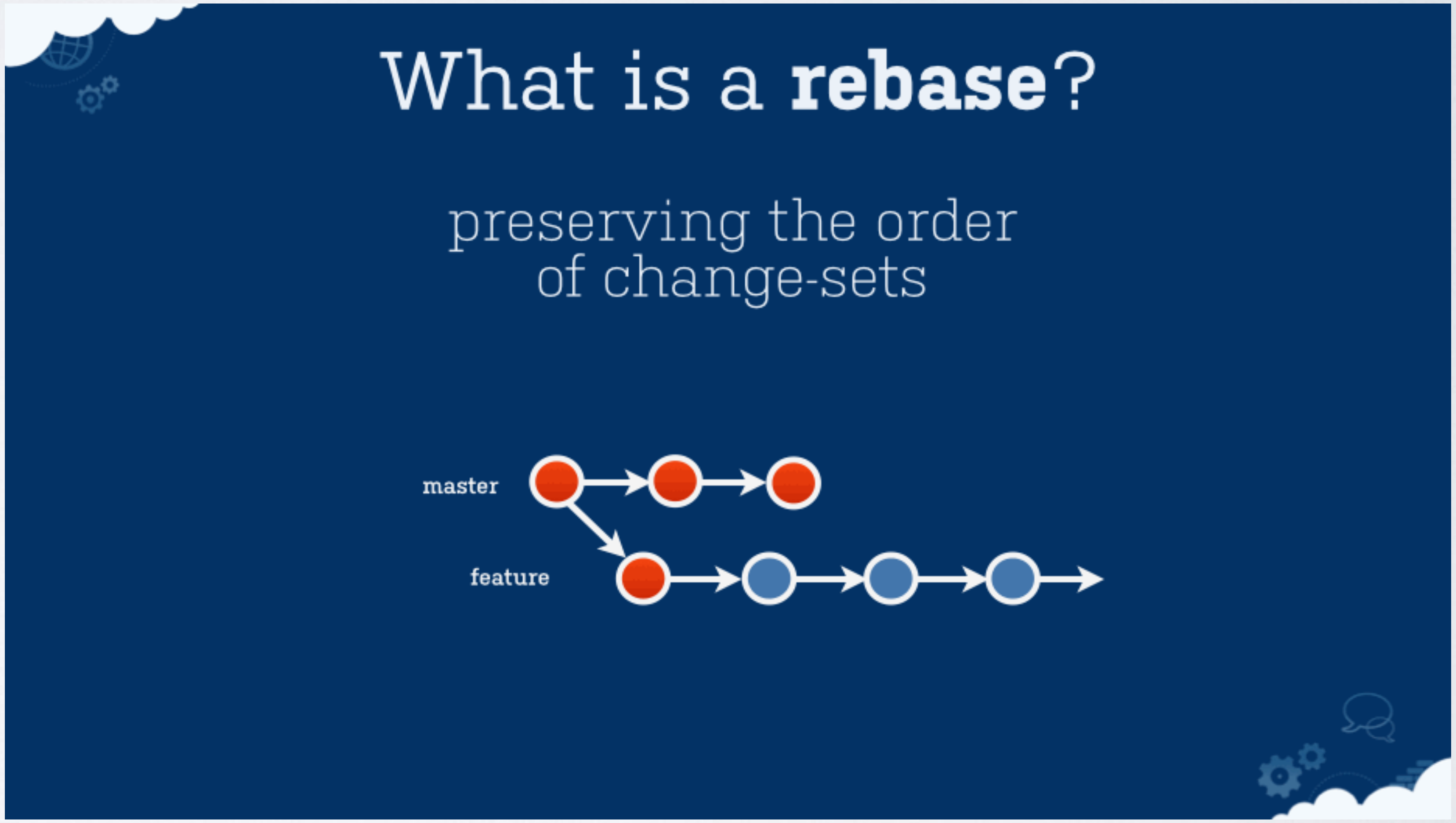
```
$ git push origin feature-1
```

then create a **merge request** online (from `feature-1` to `working-branch`)

Before merge: **CI** and **code review**

Collaboration

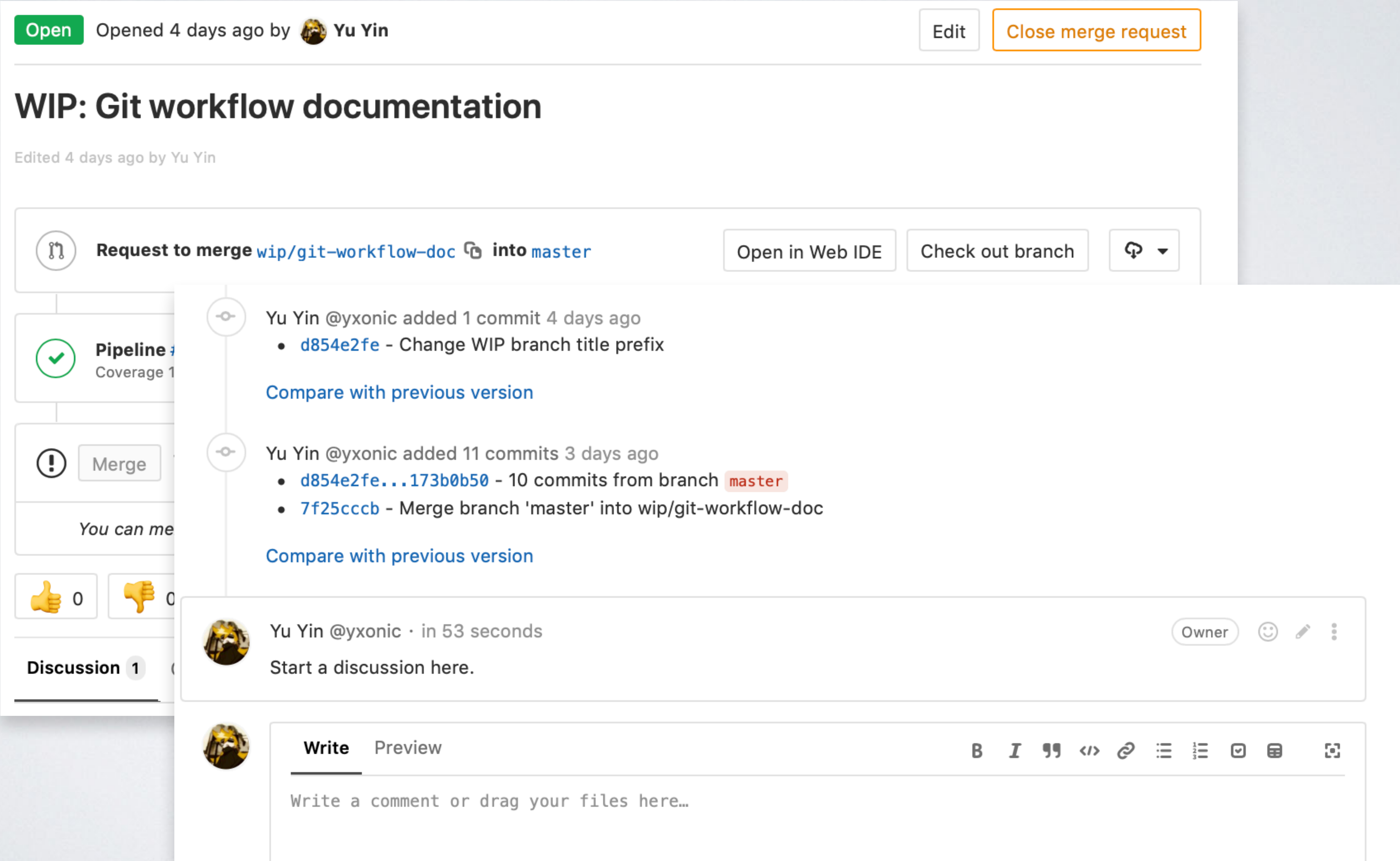




What is a **rebase**?

preserving the order
of change-sets

Collaboration



Opened 4 days ago by Yu Yin

WIP: Git workflow documentation

Edited 4 days ago by Yu Yin

Request to merge wip/git-workflow-doc into master

Pipeline #

Coverage 1

You can me

👍 0 👎 0

Yu Yin @yxonic added 1 commit 4 days ago

- d854e2fe - Change WIP branch title prefix

Compare with previous version

Yu Yin @yxonic added 11 commits 3 days ago

- d854e2fe...173b0b50 - 10 commits from branch master
- 7f25cccb - Merge branch 'master' into wip/git-workflow-doc

Compare with previous version

Yu Yin @yxonic · in 53 seconds

Start a discussion here.

Write a comment or drag your files here…

Project Management

Advanced topics for **maintainers**:

- Roles and responsibilities
- Working branch management
- Development phases
- Troubleshooting and correction

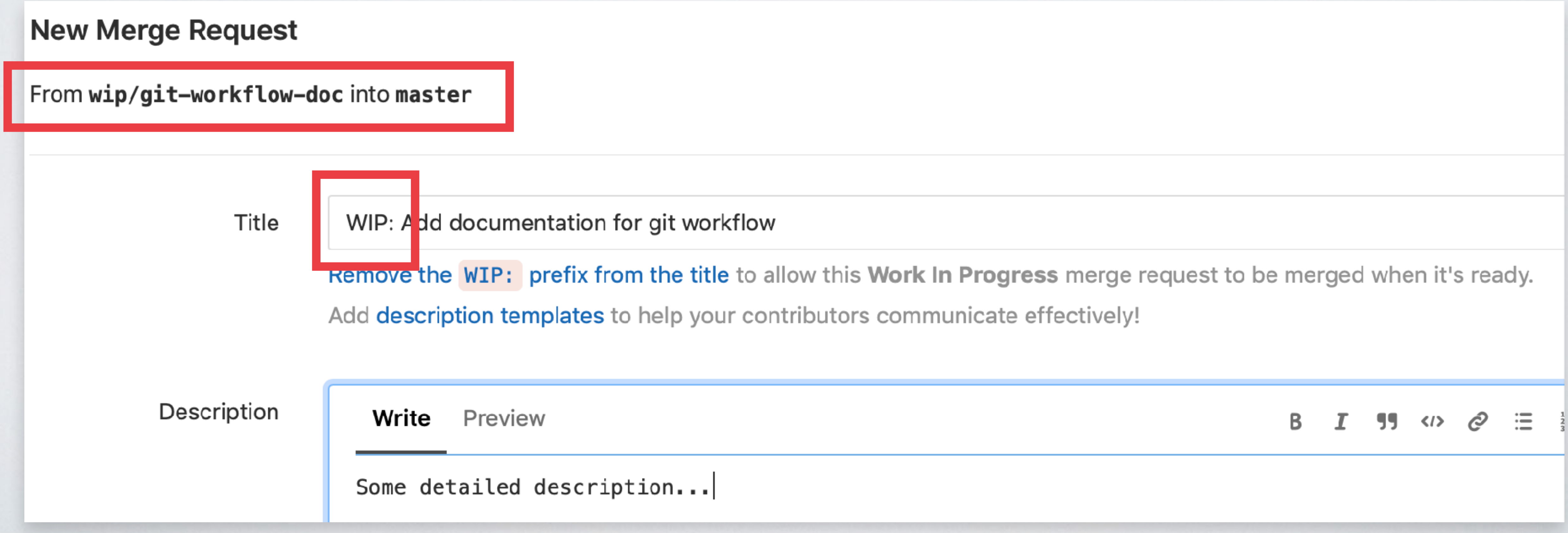
Project Management

Advanced topics for **maintainers**:

- Roles and responsibilities
- **Working branch management**
- Development phases
- Troubleshooting and correction

Working Branch Management

- Naming: `wip/some-functionality`
- Create a merge request immediately:



New Merge Request

From **`wip/git-workflow-doc`** into **`master`**

Title: WIP: Add documentation for git workflow

Remove the `WIP:` prefix from the title to allow this **Work In Progress** merge request to be merged when it's ready.

Add description templates to help your contributors communicate effectively!

Description:

```
Some detailed description...
```

Sync with other Working Branches

- **Don't** do rebase
- Sync with `master` branch frequently:

```
$ git checkout wip/working-branch

$ git fetch

$ git merge origin/master

$ git push origin wip/working-branch
```

- Pickup changes in other branches with cherry-pick

Finish a Working Branch

- Merge `master` to make this branch updated
- Pass CI tests and resolve discussions
- Remove `WIP:` from title
- Merge

Some Suggestions

- Better history with better commit messages
- Think about modularity and separation
- Always keep master branch clean and tidy

More Materials

- Git tutorial:

```
$ man gittutorial
```

- Git workflows:

  https://www.atlassian.com/git/tutorials/comparing-workflows

  https://docs.gitlab.com/ee/workflow/gitlab_flow.html

  https://www.atlassian.com/blog/archives/simple-git-workflow-simple

- BDAA workflow detailed documentation (unfinished):

  https://git.ustc.edu.cn/bdaa-kg/kg-meta/blob/wip/git-workflow-doc/developer/git-workflow.md

Q&A