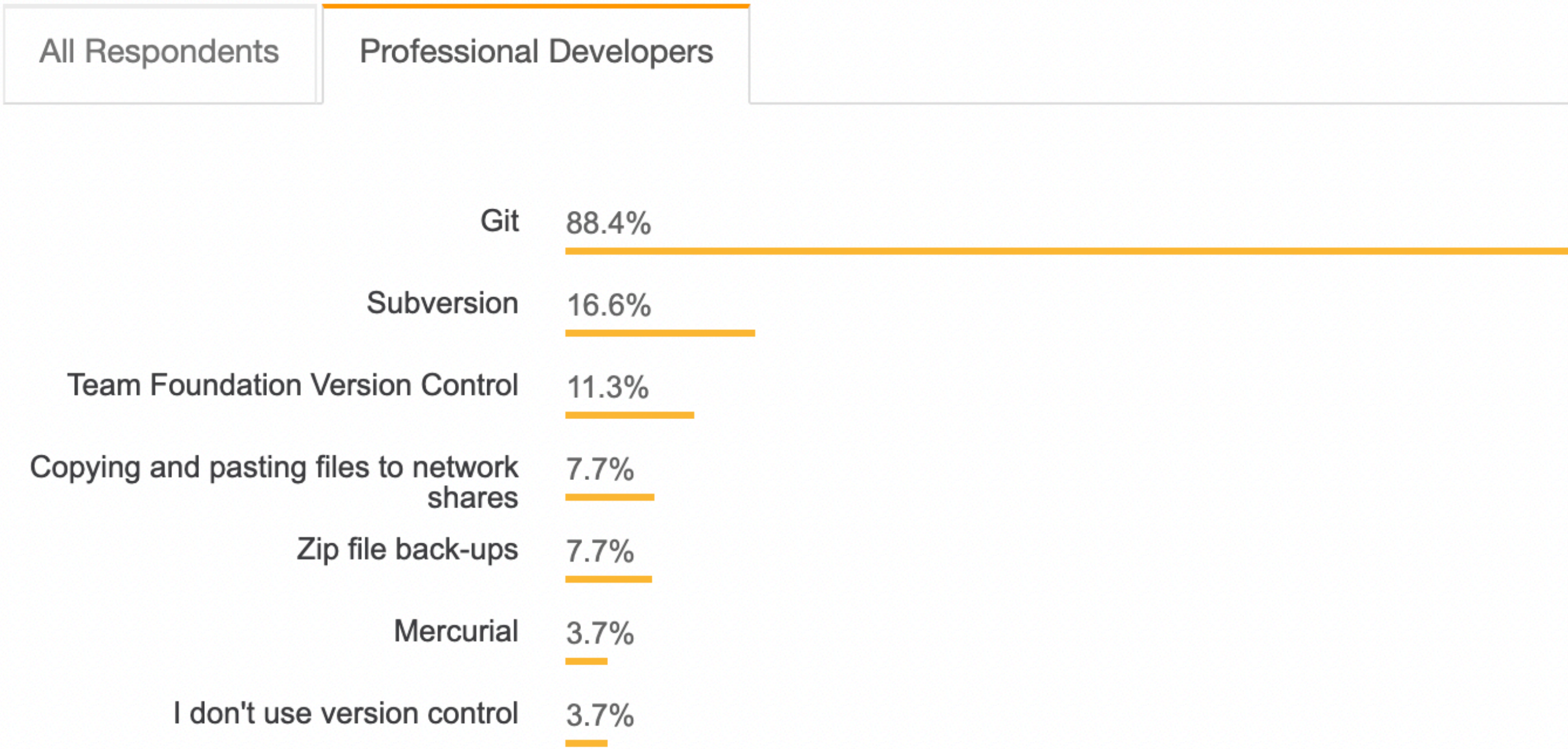


Git workflow & Git commit

youxun
03.2021

Version Control



Ref: [Stack Overflow Annual Developer Survey – 2018](#)

Note: Git is the dominant choice for version control for developers today, with almost 90% of developers checking in their code via Git.

Git workflow

- What is git workflow
- How many git workflows are there

Git branch

- What types of branches are there
- How to name the branch

Git branch

- MASTER branch
- FEATURE branch
- DEVELOP branch
- RELEASE branch
- HOTFIX branch

Branching workflows

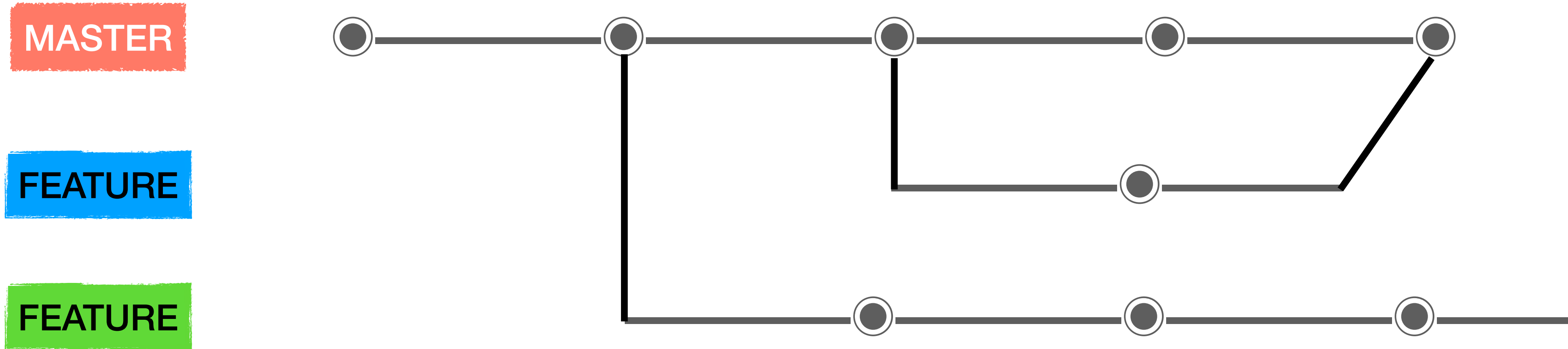
- Basic Git Workflow
- Git Feature Branch Workflow
- Git Feature Workflow with Develop Branch
- Git Fork Workflow
- Gitflow Workflow
- GitHub Workflow
- GitLab Workflow

Basic Git Workflow

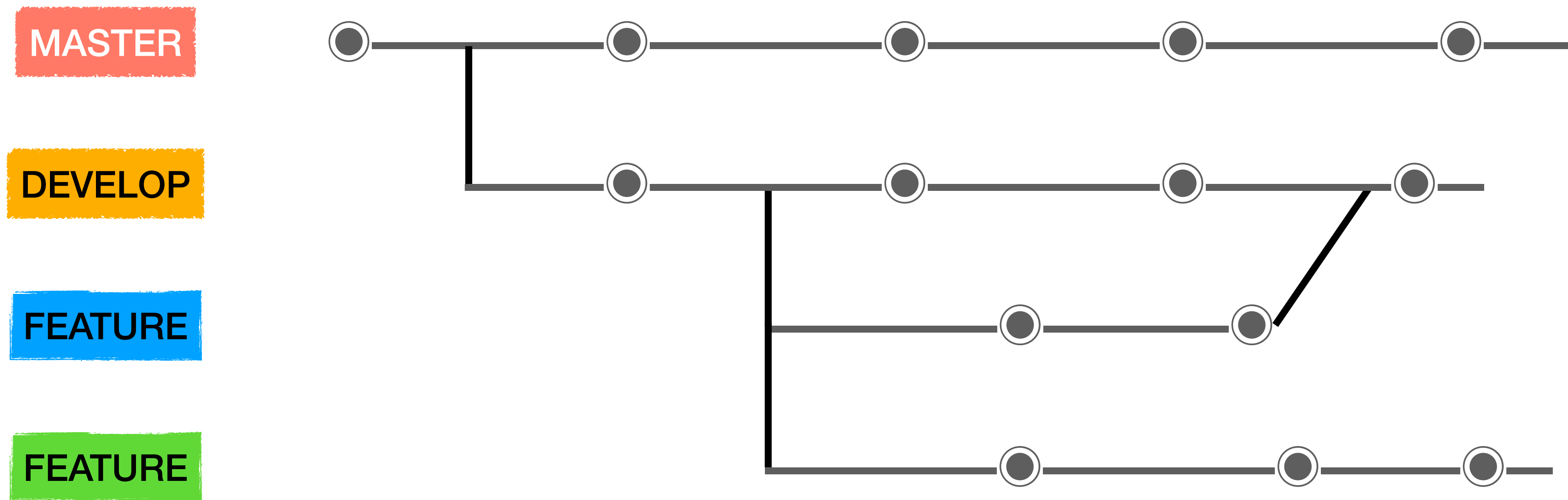
Centralized Workflow



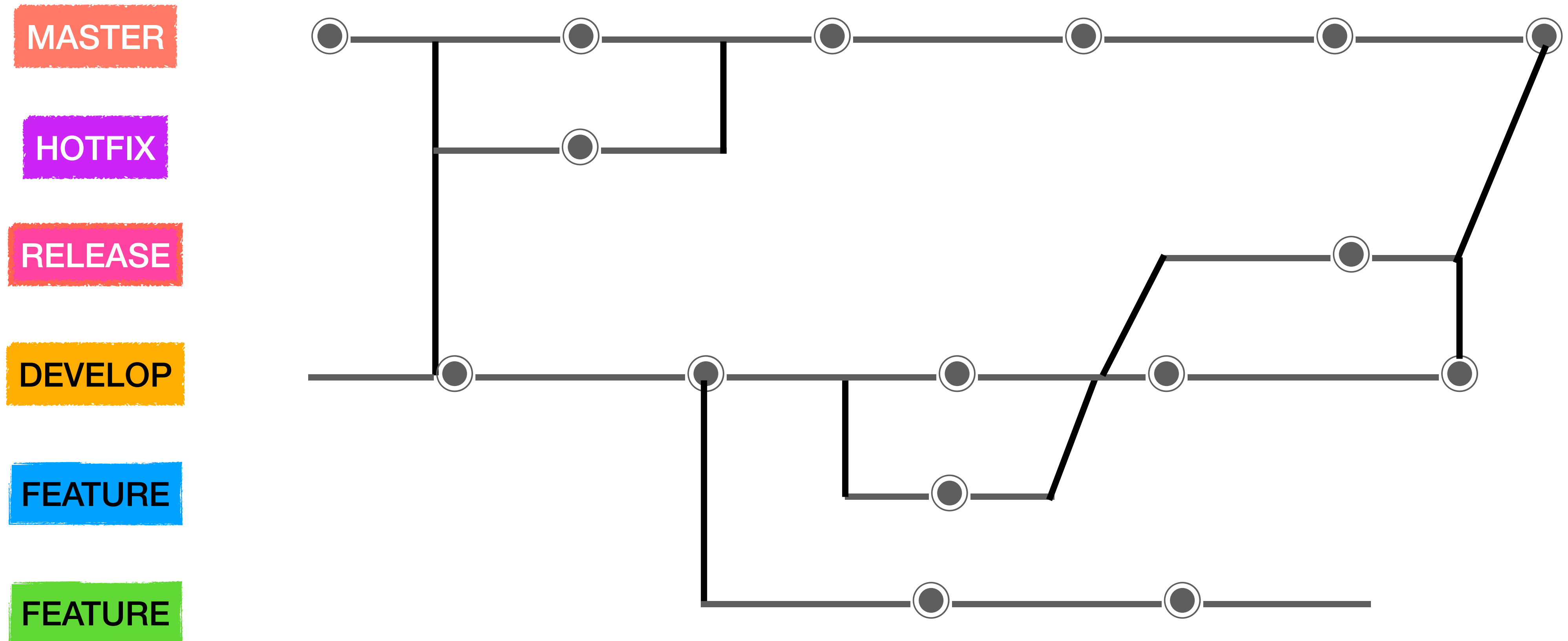
Git Feature Branch Workflow



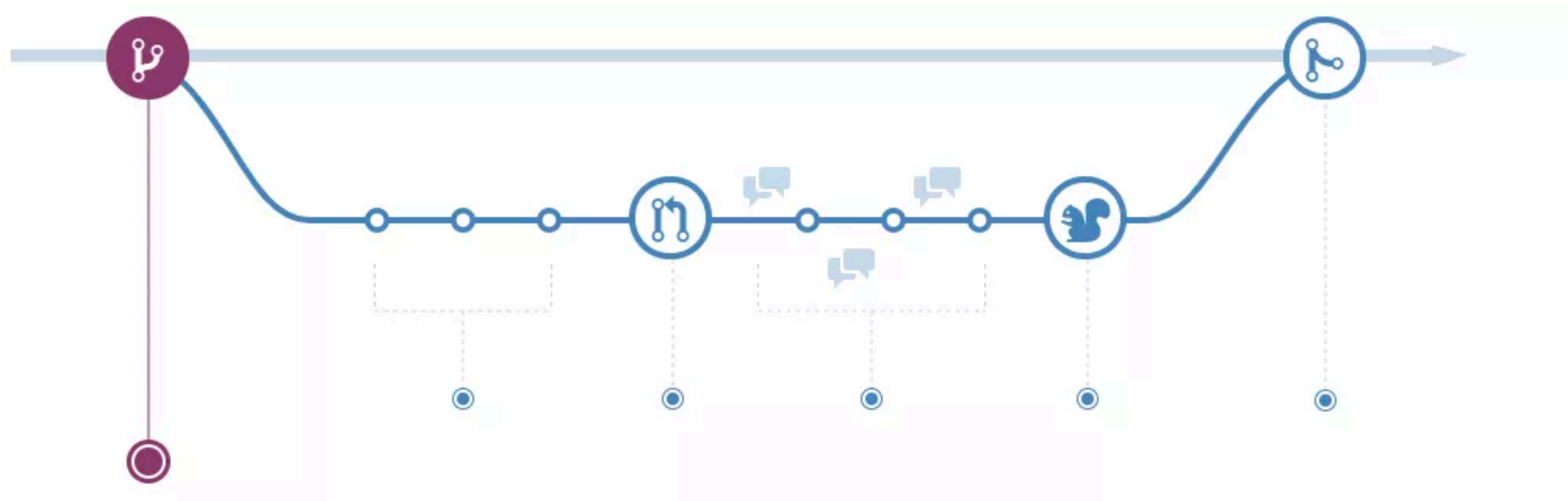
Git Feature Workflow with Develop Branch



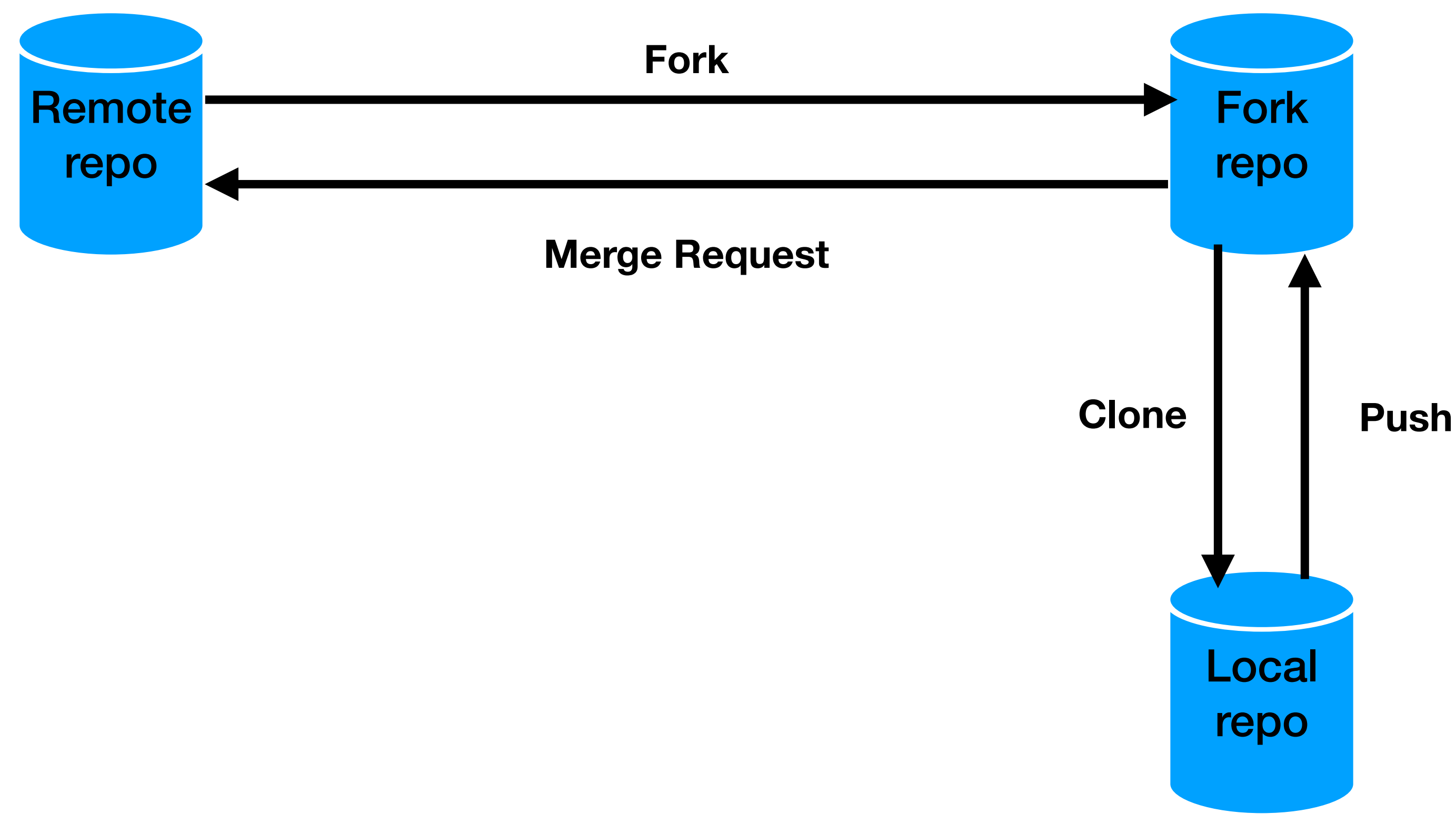
Gitflow Workflow



GitHub Workflow



Git Fork Workflow



Git workflow

- Advantages
- Disadvantages
- Conflict
- Code / Environment

Conflicts

Merge conflicts can be an intimidating experience. Luckily, Git offers powerful tools to help navigate and resolve conflicts. Git can handle most merges on its own with automatic merging features. A conflict arises when two separate branches have made edits to the same line in a file, or when a file has been deleted in one branch but edited in the other. Conflicts will most likely happen when working in a team environment.

Git commit

- What types of commits are there
- How to write a commit log
- Advantages

What types of commits are there

- FEAT
- FIX
- DOCS
- STYLE
- REFACTOR
- TEST
- CHORE

Git branch naming conventions

How to name the branch


















- <AUTHOR>-<BRANCH-TYPE>-<DESC>
 - Use hyphens as separators
- OTHER

How to write a commit log

- <type>[optional scope]: <description>
 - [optional body]
 - [optional footer(s)]
- <[type] | [issue-tracker-ID]>: <description>

Ref: <https://www.conventionalcommits.org/en/v1.0.0/>

The example of commit

github.com/golang/go		
 .github	.github: add link to questions in ISSUE_TEMPLATE	14 months ago
 api	syscall: restore broken GetQueuedCompletionStatus signature but m...	17 days ago
 doc	cmd/go: remove -insecure flag on go get	11 days ago
 lib/time	lib/time, time/tzdata: update tzdata to 2021a	2 months ago
 misc	all: use HTML5 br tags	13 hours ago
 src	all: remove duplicate words	4 hours ago
 test	cmd/compile: test register ABI for method, interface, closure calls	19 hours ago
 .gitattributes	all: treat all files as binary, but check in .bat with CRLF	9 months ago
 .gitignore	.gitignore: ignore src/cmd/dist/dist	3 years ago
 AUTHORS	A+C: add new e-mail addresses for Andy Pan	2 days ago
 CONTRIBUTING.md	all: restore changes from faulty merge/revert	3 years ago
 CONTRIBUTORS	A+C: add new e-mail addresses for Andy Pan	2 days ago
 LICENSE	doc: revert copyright date to 2009	5 years ago
 PATENTS	LICENSE: separate, change PATENTS text	10 years ago
 README.md	README: pull gopher image from website	25 days ago
 SECURITY.md	SECURITY.md: update go versions	2 years ago
 codereview.cfg	codereview.cfg: add codereview.cfg for master branch	22 days ago

Linus

I also refuse to pull commits that have been made with the github web interface. Again, the reason for that is that the way the github web interface work, those commits are invariably pure crap. Commits done on github invariably have totally unreadable descriptions, because the github commit making thing doesn't do *any* of the simplest things that the kernel people expect from a commit message:

- no "short one-line description in the first line"
 - no sane word-wrap of the long description you type: github commit messages tend to be (if they have any description at all) one long unreadable line.
 - no sign-offs etc that we require for kernel submissions.
- Extension
 - <https://github.com/thoughtbot/dotfiles/blob/master/gitmessage>
 - <https://gitmoji.dev/>

**What is a successful Git workflow
for your team?**

Thanks