# Retrospective
## Git assignement

18/10/2016
Cécile Camillieri

# Marking

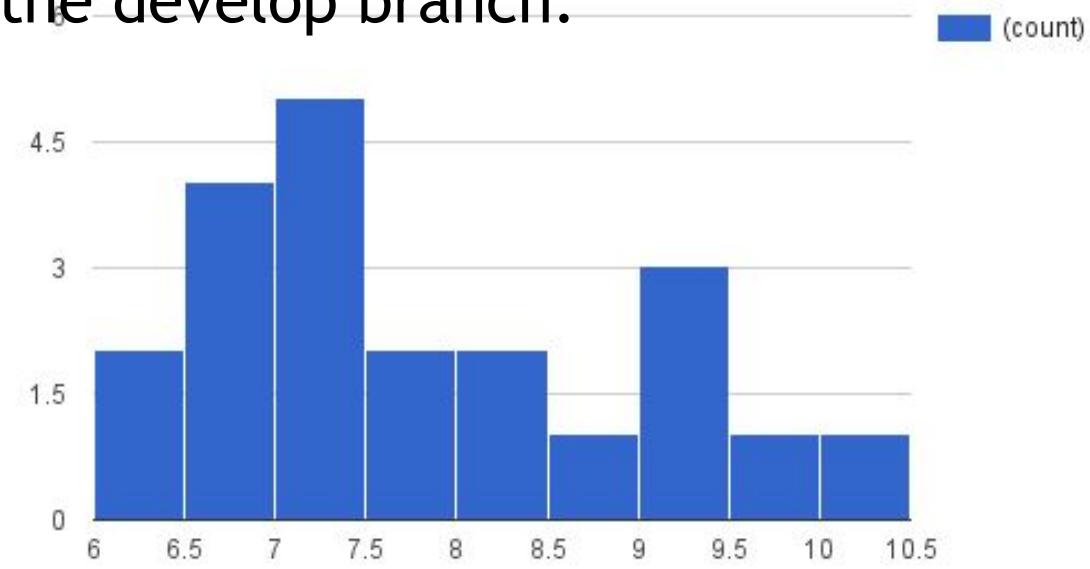
- Email and name properly set /1
- Commit messages quality /1
- Commit with menu.txt /1
- Commit of log.txt /1
- Commit of diff.txt /1
- Merge of develop in master /1
- Tag v1.0 /1
- Progress in git flow part /1.5
- Develop pushed /1
- Tag v2.0 (at correct location) /0.5

- -1 if not on Github Classroom
- -1 if late

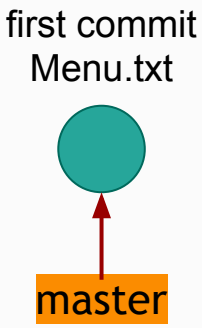
# Some statistics

- 20 repositories
- 2 of them not on Github Classroom
- 18 have tagged v1.0 with expected history
- 9 have a v2.0 tag. Only 4 with expected history.
- Only 5 have pushed the develop branch.

- Average mark: 7,57/10
- Minimum: 6,5
- Maximum: 10

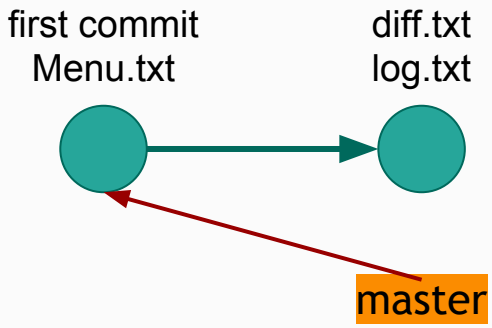


# Expected repository



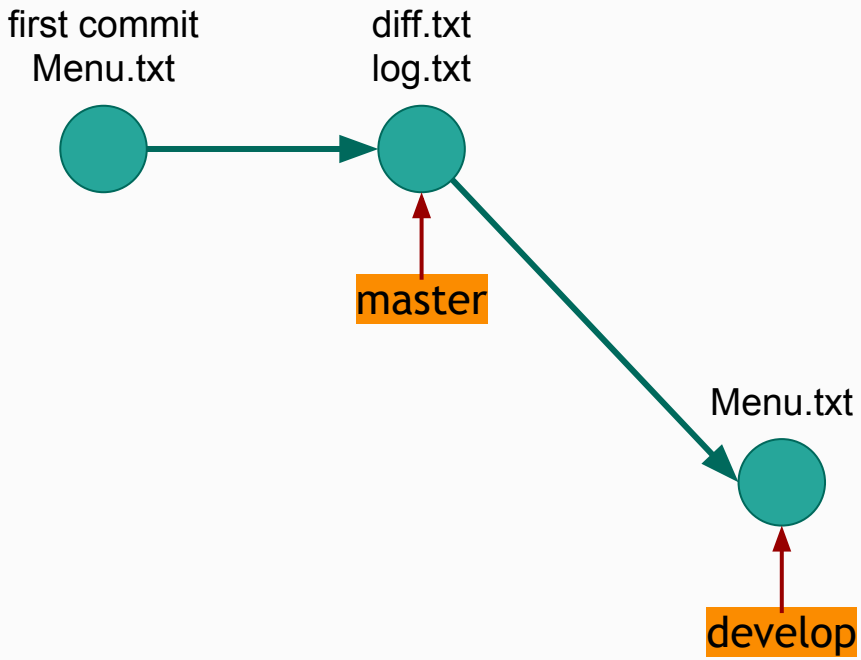

```
git add Menu.txt
git commit -m "create menu."
```

# Expected repository



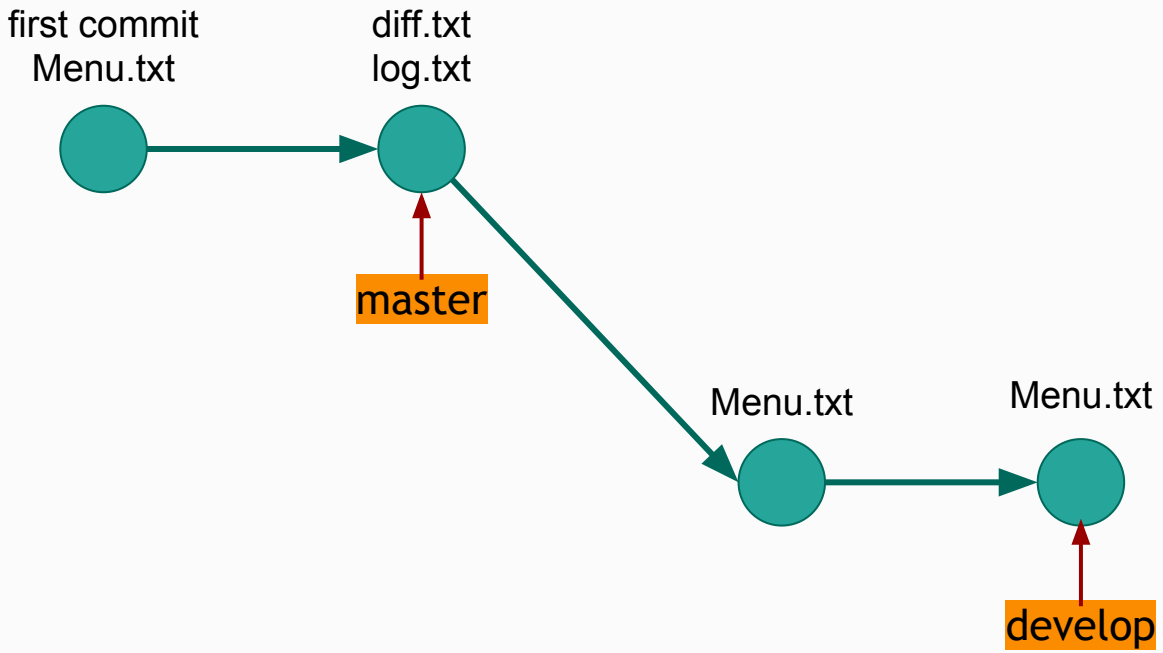

```
git log > log.txt
git diff commitId > diff.txt
git add --all
git commit -m "log and diff file"
```

# Expected repository



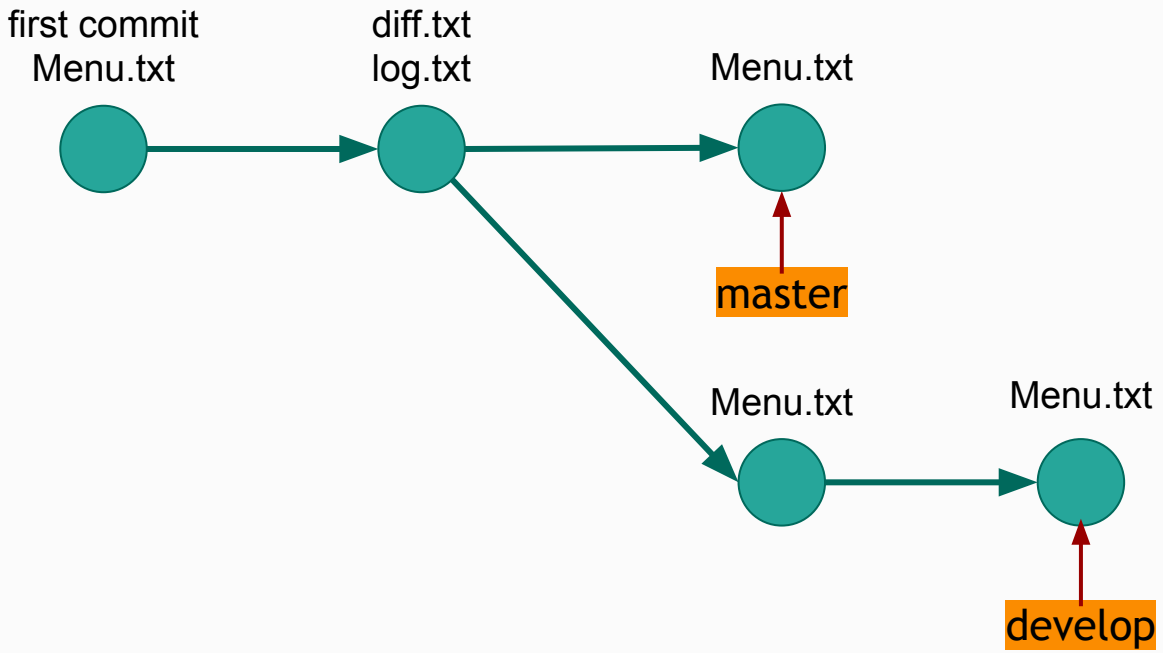

```
git checkout -b develop
git add Menu.txt
git commit -m "modif menu on develop"
```

# Expected repository



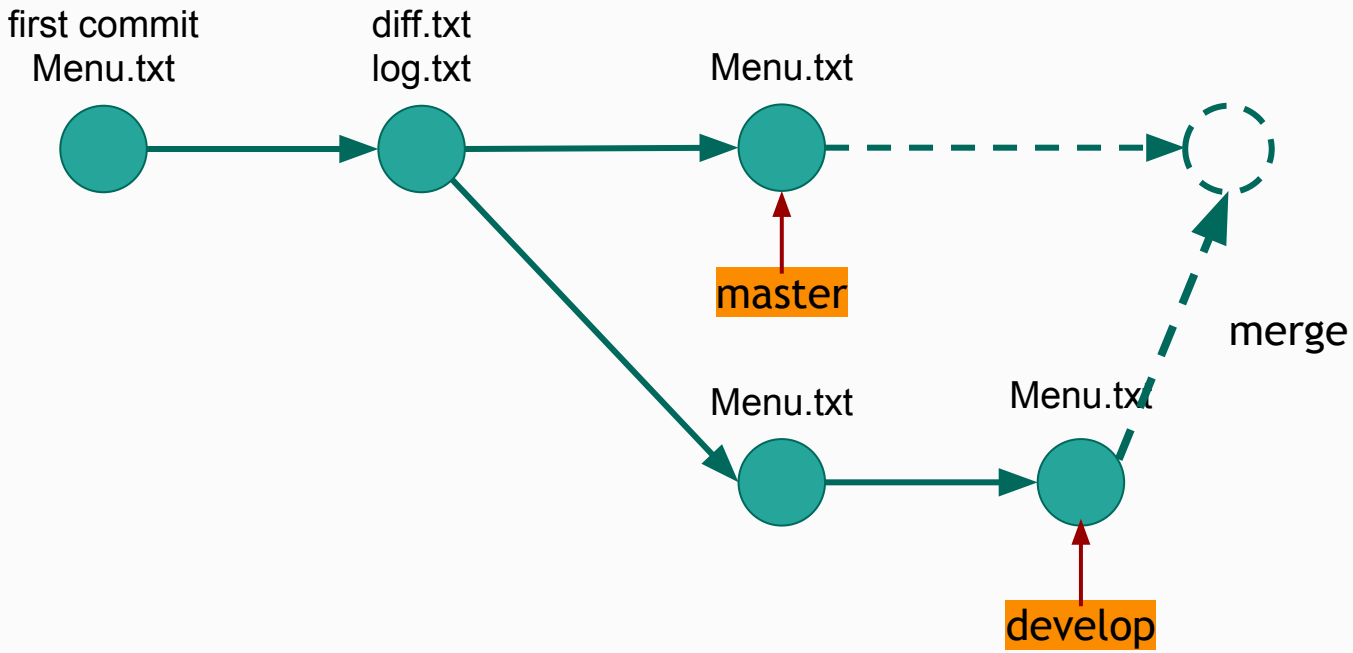

```
git add Menu.txt
git commit -m "modif menu on develop"
```

# Expected repository



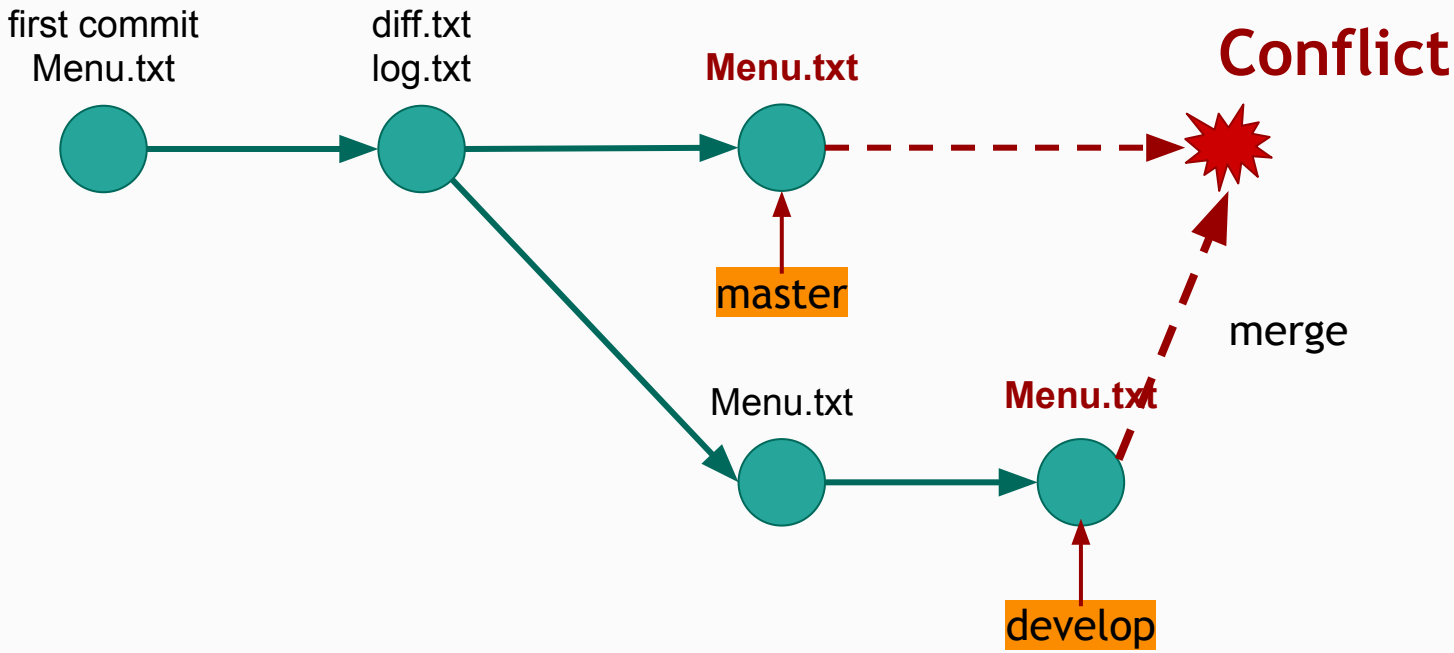

```
git checkout master
git add Menu.txt
git commit -m "modif menu on master"
```

# Expected repository

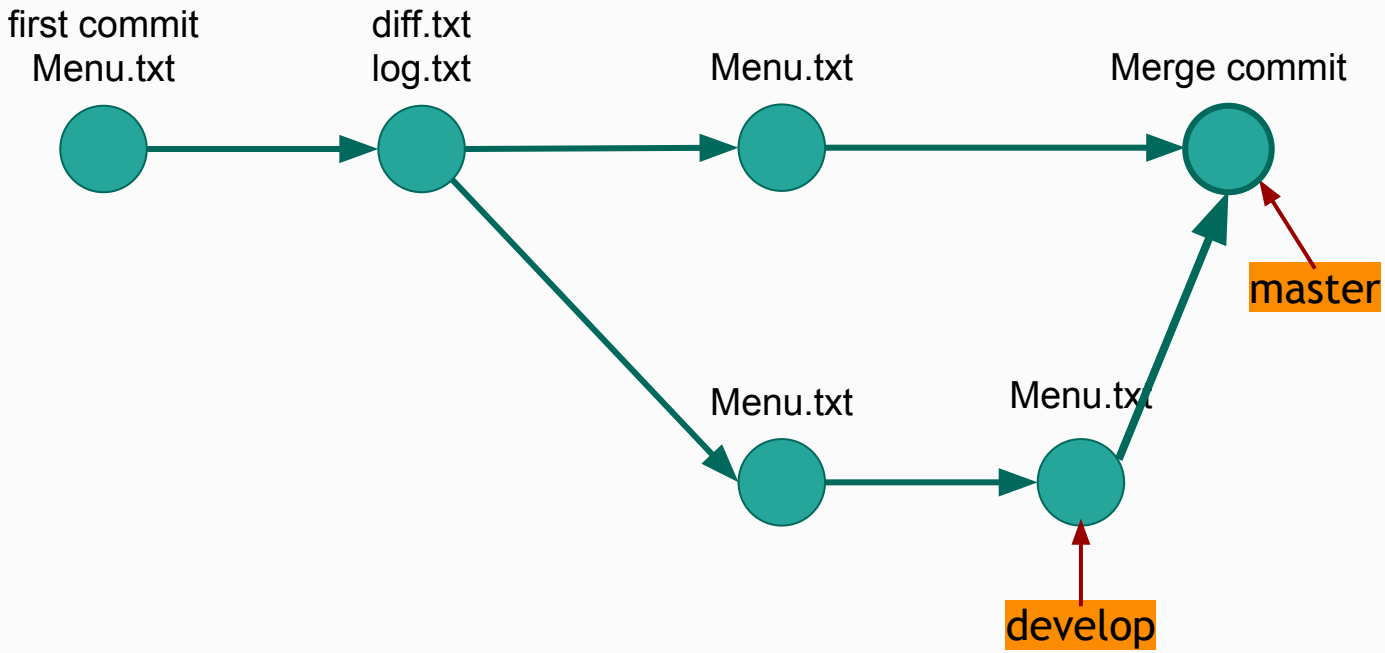



```
git merge develop
```

# Expected repository

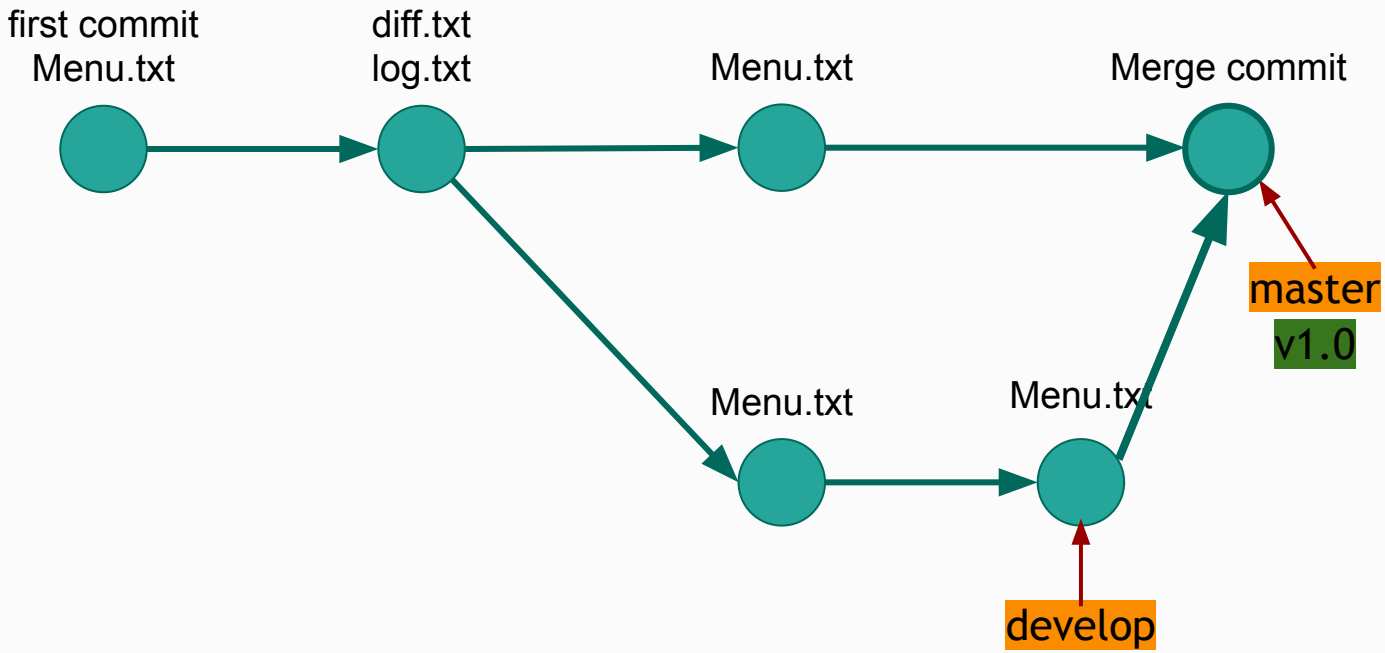


# Expected repository



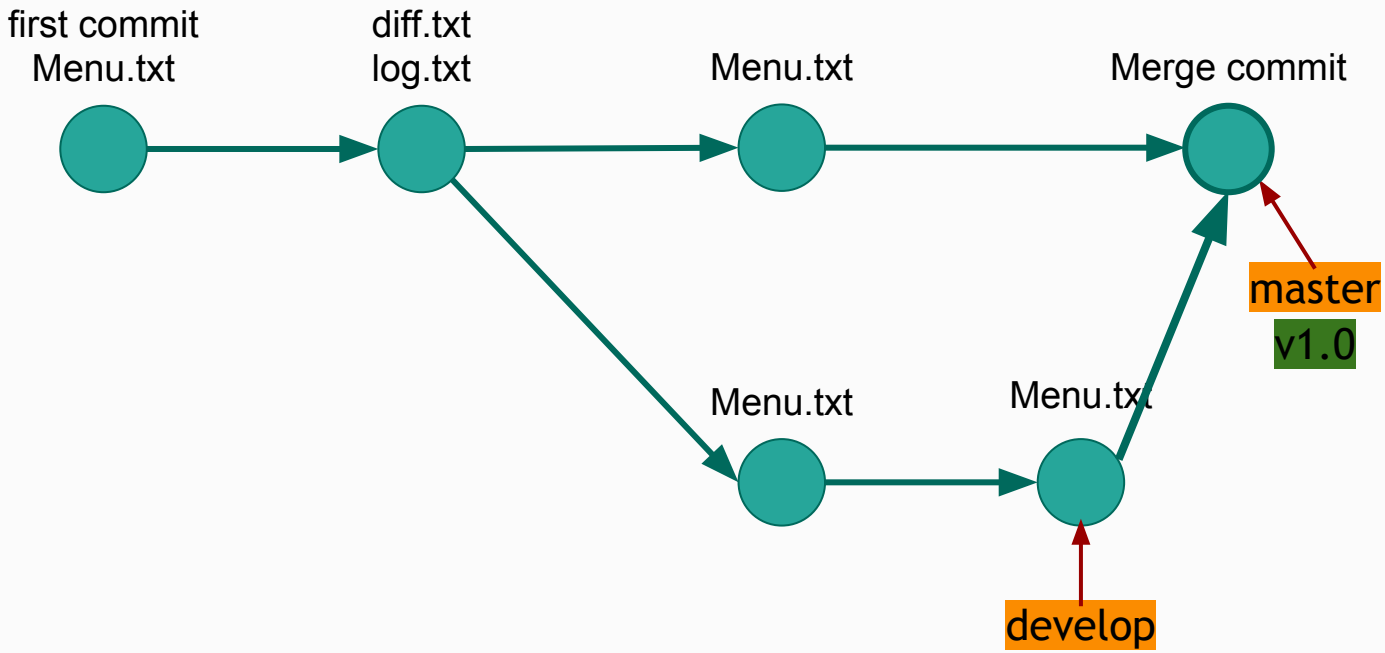

```
git add Menu.txt
git commit -m "merge develop on master"
```

# Expected repository



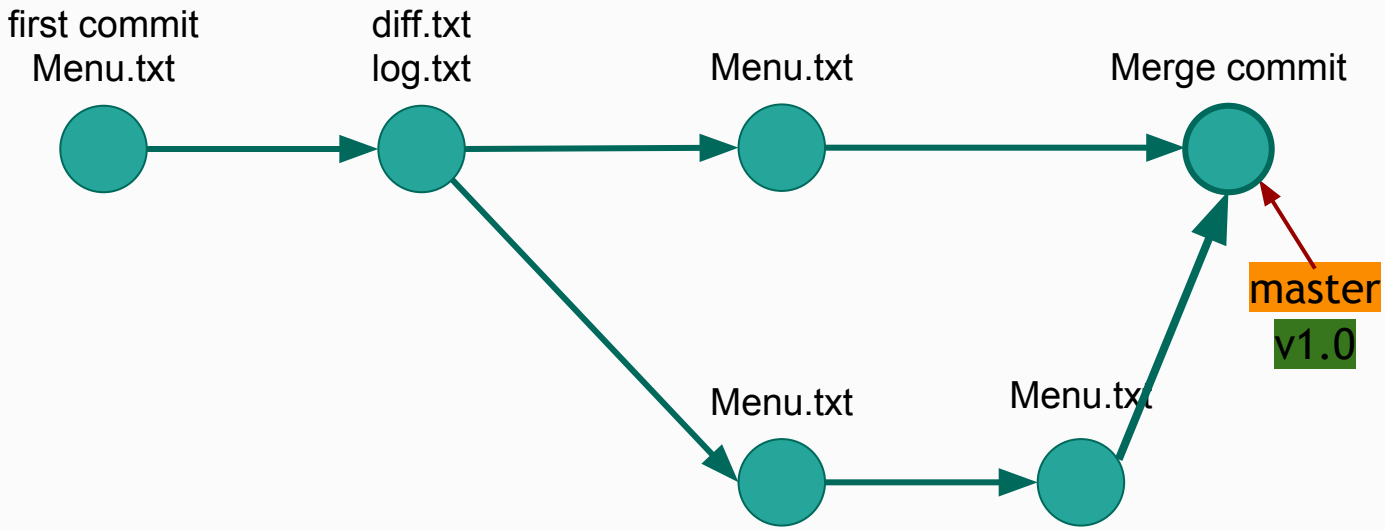

```
git tag -a v1.0 -m "version 1.0"
```

# Expected repository

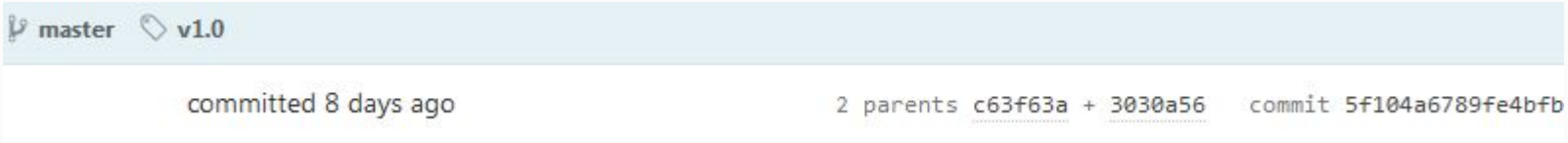



```
git push origin master
git push origin v1.0
```

# Expected repository - remote




- We only pushed master
- Remote doesn't know about develop
- But it knows that the merge commit has 2 "parents"



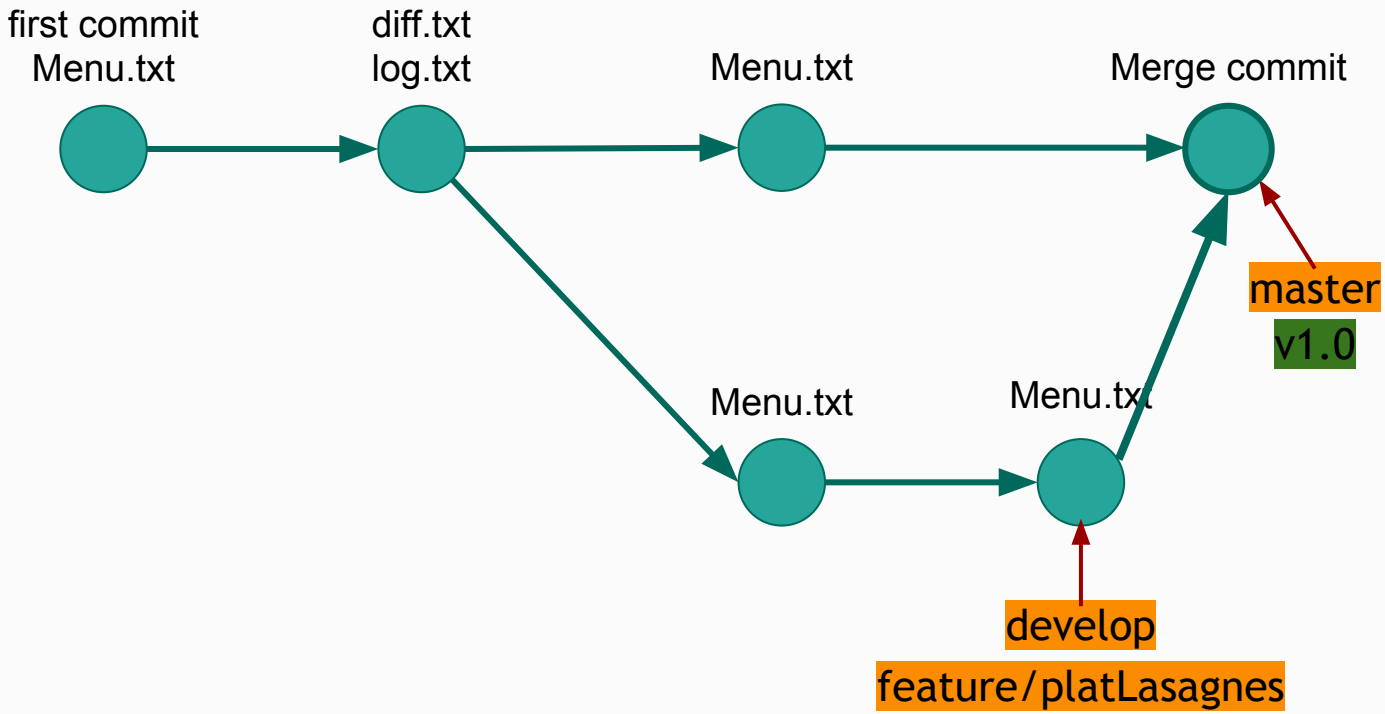
# Git flow feature start

```
git flow feature start platLasagnes
```

```
git checkout develop
git branch feature/platLasagnes
git checkout feature/platLasagnes
```

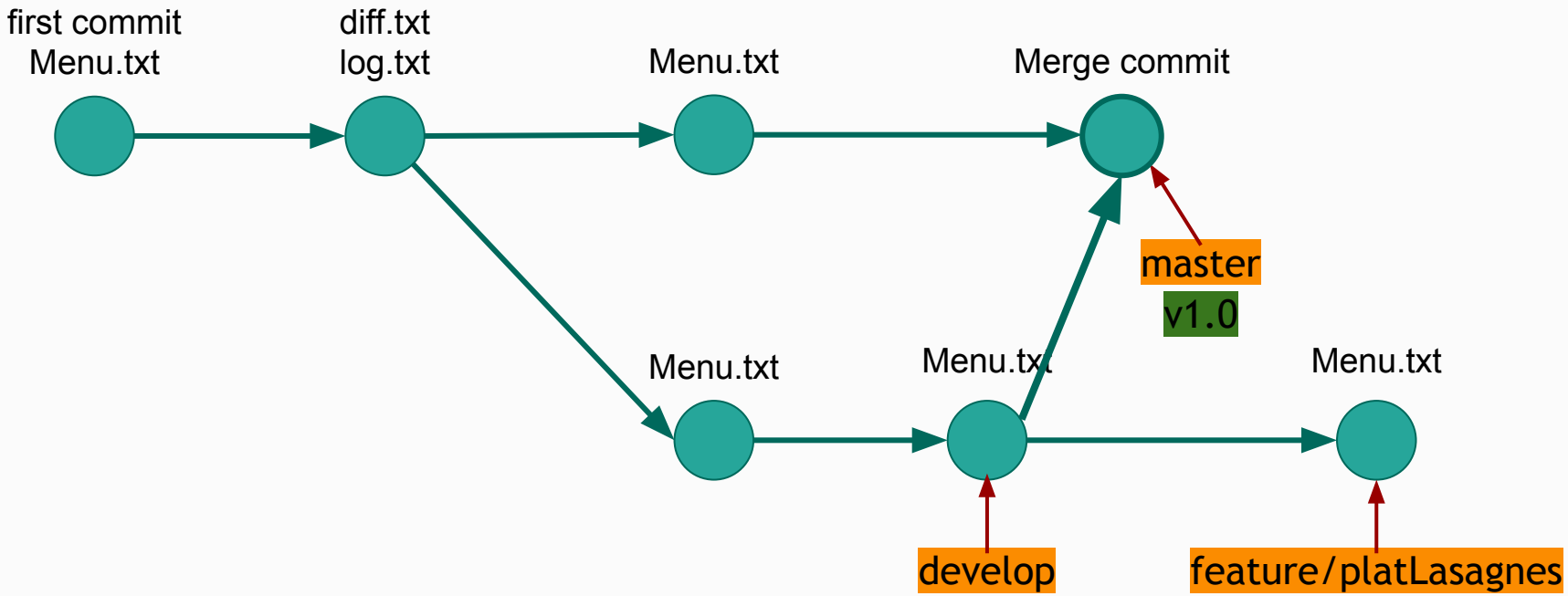
```
$ git flow feature start platLasagnes
Switched to a new branch 'feature/platLasagnes'

Summary of actions:
- A new branch 'feature/platLasagnes' was created, based on 'develop'
- You are now on branch 'feature/platLasagnes'

Now, start committing on your feature. When done, use:

     git flow feature finish platLasagnes
```

# Expected repository - local



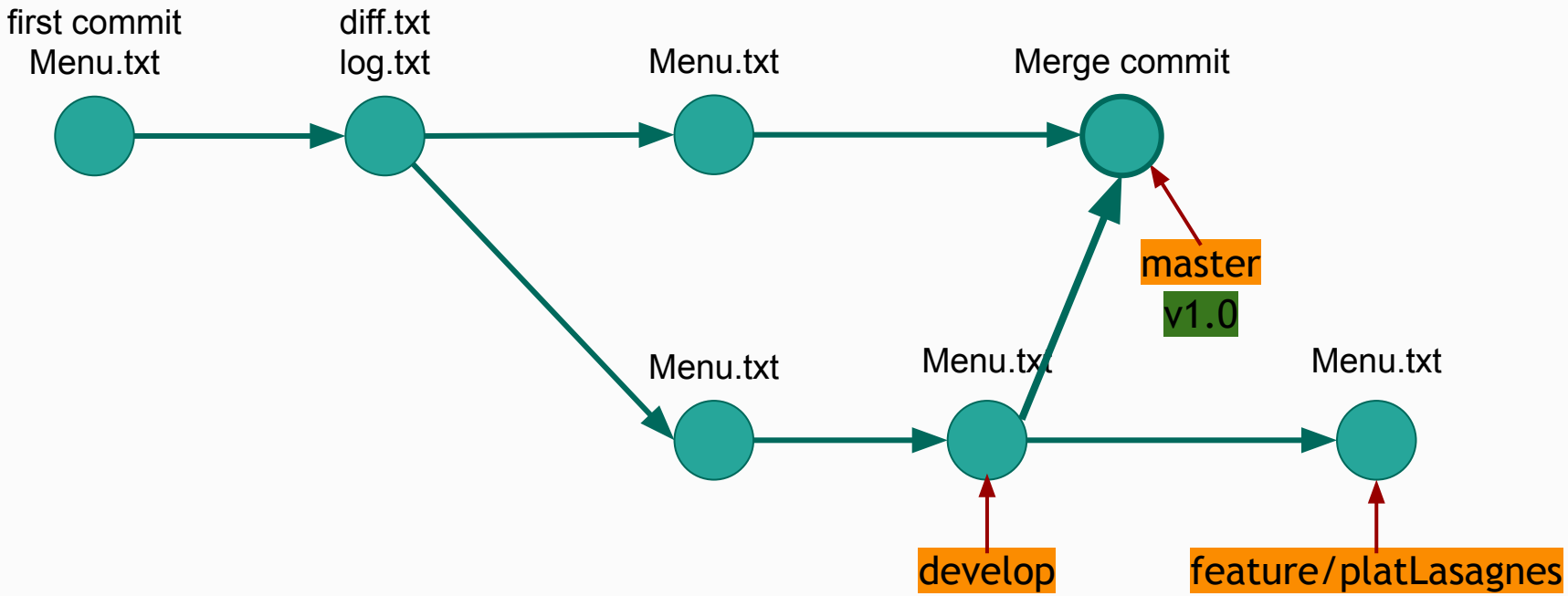

- Features start from **develop**
- Does not know about changes from master
- Should have merged master in develop as well.

# Expected repository - local




```
git add Menu.txt
git commit -m "add lasagnas in the menu"
```

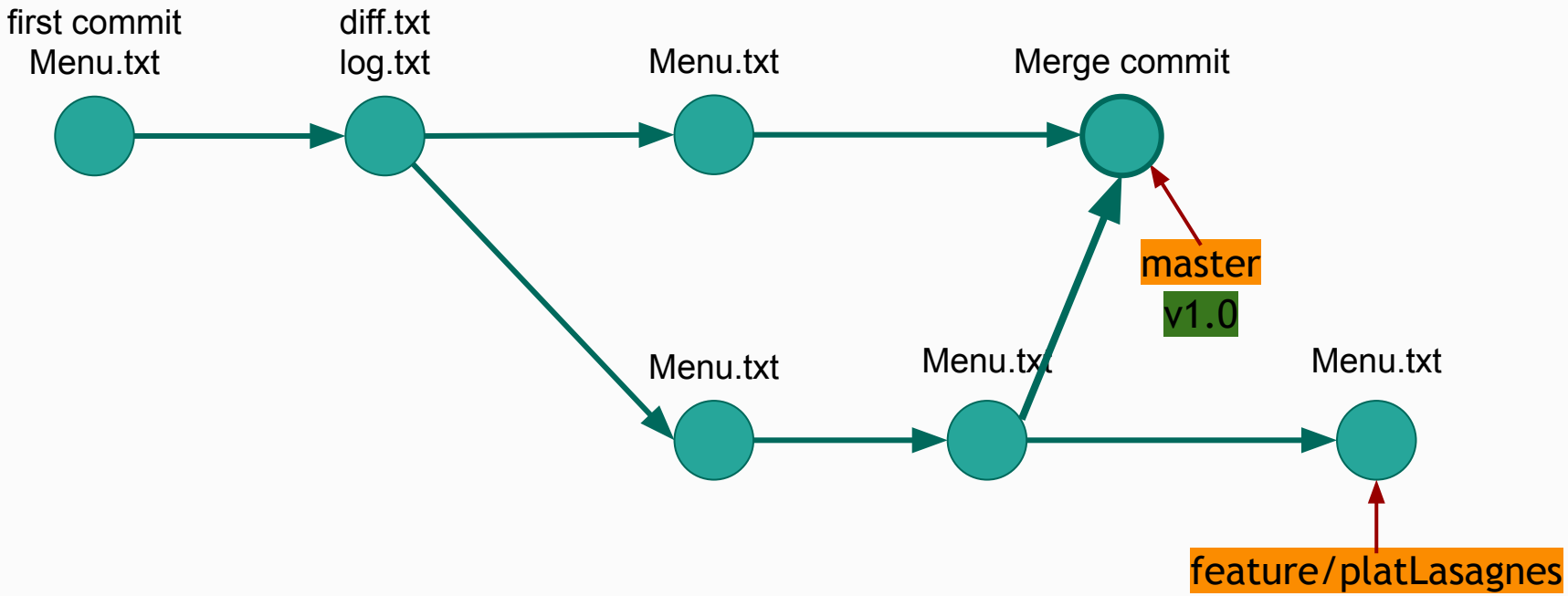
# Expected repository - local




**`git push`** -> does not work

**`git push`** `origin feature/platLasagnes`

# Expected repository - remote




- We only pushed the feature's branch
- Remote still doesn't know about develop

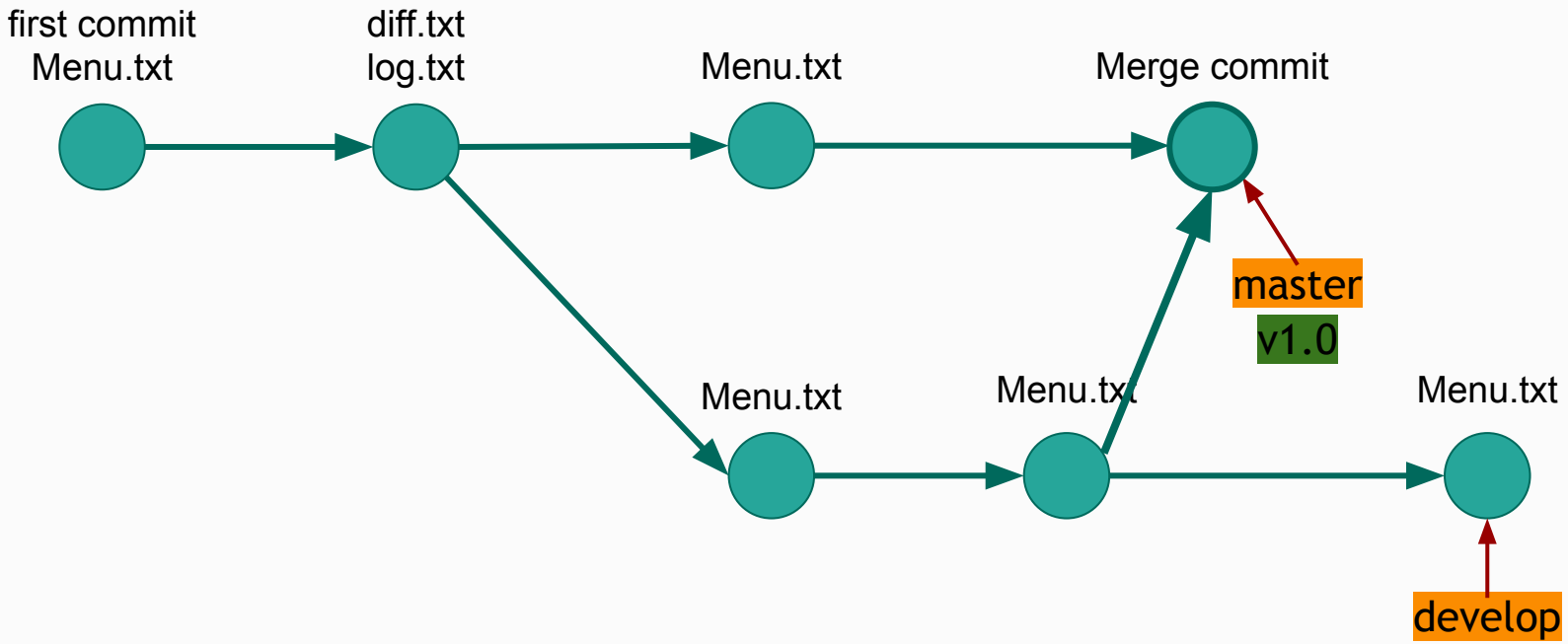
# Git flow feature finish

```
git flow feature finish platLasagnes
```

```
git checkout develop
git merge feature/platLasagnes
git branch -d feature/platLasagnes
```

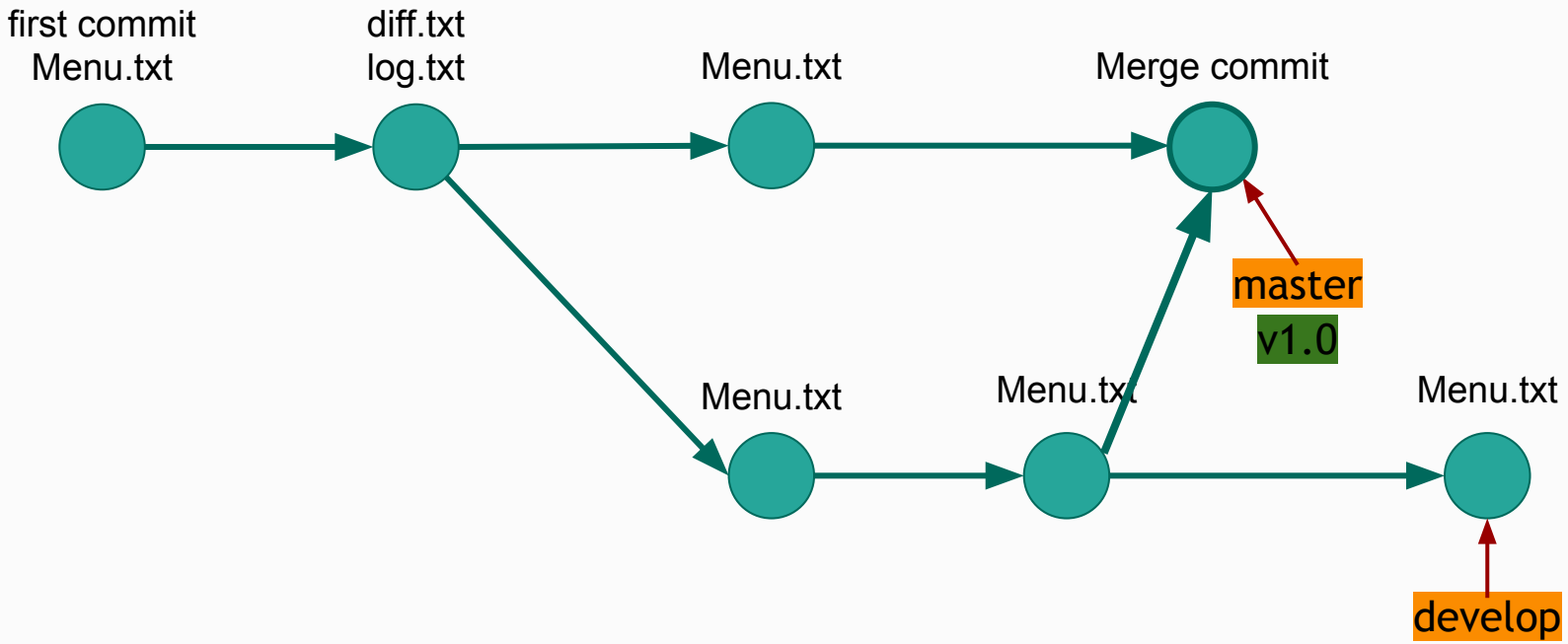
```
$ git flow feature finish platLasagnes
Switched to branch 'develop'
Updating 8e14aee..d8e42fe
Fast-forward
 Menu.txt | 1 +
 1 file changed, 1 insertion(+)
 create mode 100644 Menu.txt
Deleted branch feature/platLasagnes (was d8e42fe).

Summary of actions:
- The feature branch 'feature/platLasagnes' was merged into 'develop'
- Feature branch 'feature/platLasagnes' has been locally deleted
- You are now on branch 'develop'
```

# Expected repository - local




- No changes in develop before merging
- => No merge commit
- Branche feature/platLasagnes was deleted

# Expected repository - remote



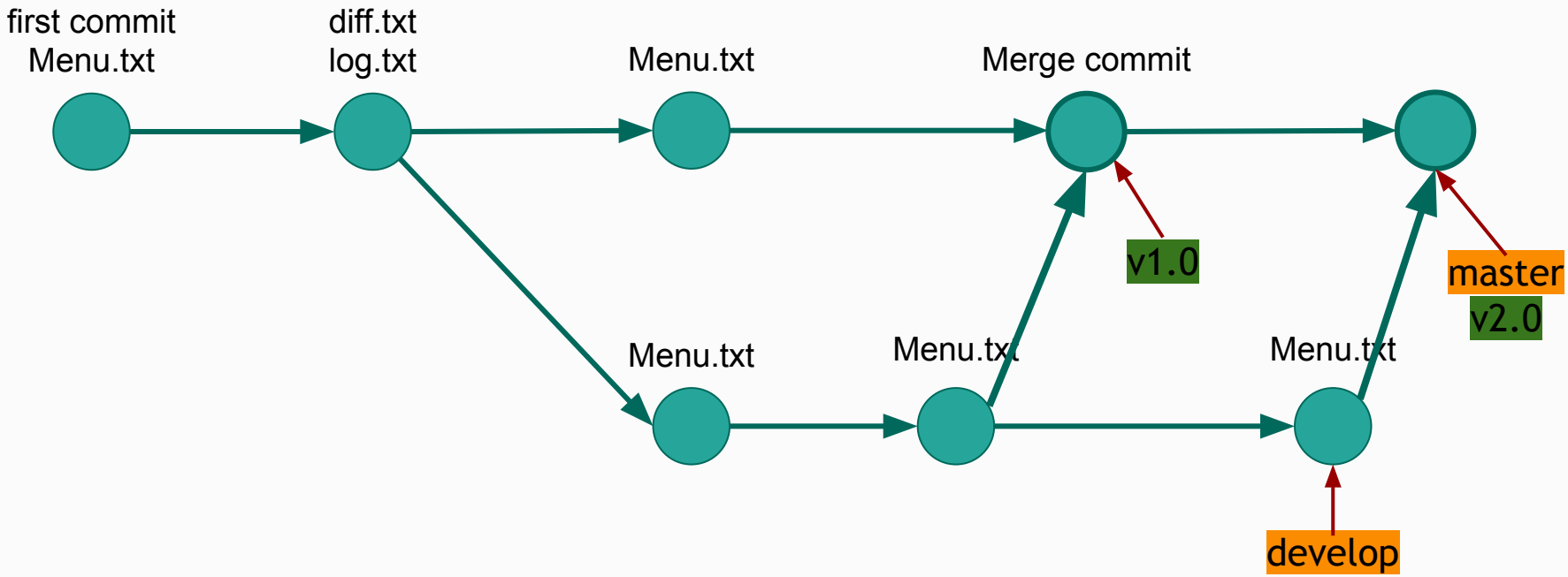

```
git push origin develop
```

# Git flow release

**git flow release start** v2.0
**git flow release finish** v2.0

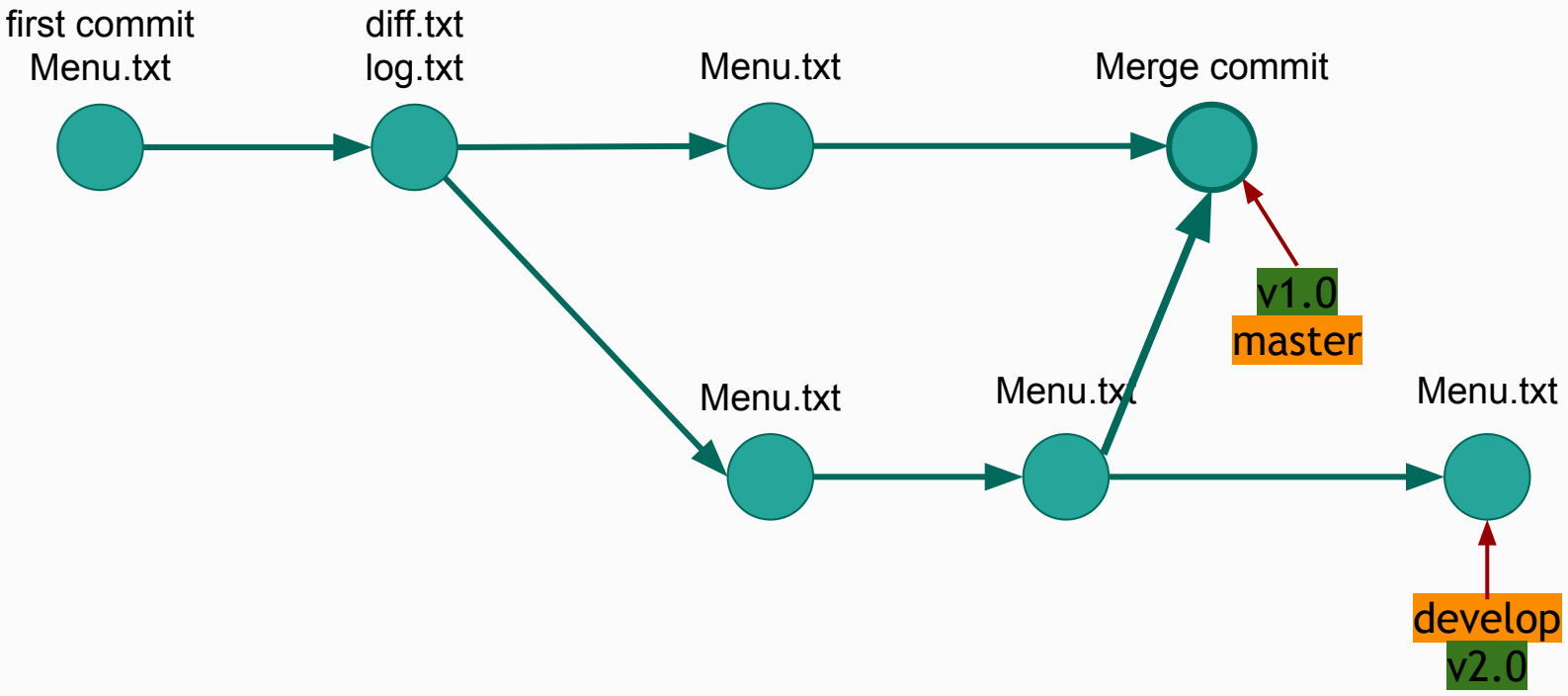
```
glc@Stagiaire-PC MINGW64 ~/git/LP/git101/newrepo (develop)
$ git flow release start v2
Switched to a new branch 'release/v2'

Summary of actions:
- A new branch 'release/v2' was created, based on 'develop'
- You are now on branch 'release/v2'

Follow-up actions:
- Bump the version number now!
- Start committing last-minute fixes in preparing your release
- When done, run:

    git flow release finish 'v2'

glc@Stagiaire-PC MINGW64 ~/git/LP/git101/newrepo (master)
$ git flow release finish v2
Switched to branch 'develop'
Merge made by the 'recursive' strategy.
 Menu.txt | 2 +-
 1 file changed, 1 insertion(+), 1 deletion(-)
Deleted branch release/v2 (was f215aa5).

Summary of actions:
- Release branch 'release/v2' has been merged into 'master'
- The release was tagged 'v2'
- Release tag 'v2' has been back-merged into 'develop'
- Release branch 'release/v2' has been locally deleted
- You are now on branch 'develop'
```

# Expected repository - final



# Except...

# Expected repository - final



# Common mistakes

- Unclear commit messages
  -> Hard to know where you stopped.
- Develop not pushed
  -> Even harder to know where you stopped.
- Missing tag(s)
  -> Can’t do anything for you…
- Git flow was not used
  -> You made something but not what was asked...

# My guess on the reasons

- Skipped lines/questions in the subject…

- Not reading what's displayed in the console
  -> You don't know the command didn't work
  -> You don't know why

Don't forget to use **git status** and **git log** to make sure your commands went through correctly

# Sprint-0

# Some more statistics

- 6 repositories
- None did all that was asked
- 6 pushed the tag properly and on time
- 5 (almost) properly updated the readme
- 5 added the ws-shapes folder
- Only one had the whole team join the repository

  => 5 people did not join their team

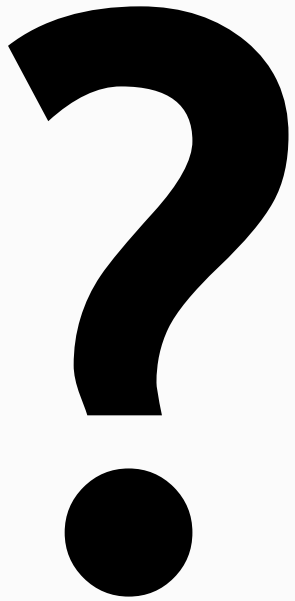
# Time to wake up!

# Conclusion

# Best practices: reminder

- **Commit** often, small.
- Always add **understandable** **commit messages.**
- **Reference Jira tasks** in commit messages.
- **.gitignore:** don’t commit unnecessary files.
- Don’t forget to **add** files after resolving merge conflicts.
- Don’t forget to **push** branches and tags
- Use **branches** (at least the master/develop model).

# But above all...

*Please*...

Read. Properly. Everything.