

# Retrospective

## Git assignement & Sprint 0

12/12/2016

Cécile Camillieri



# Marking

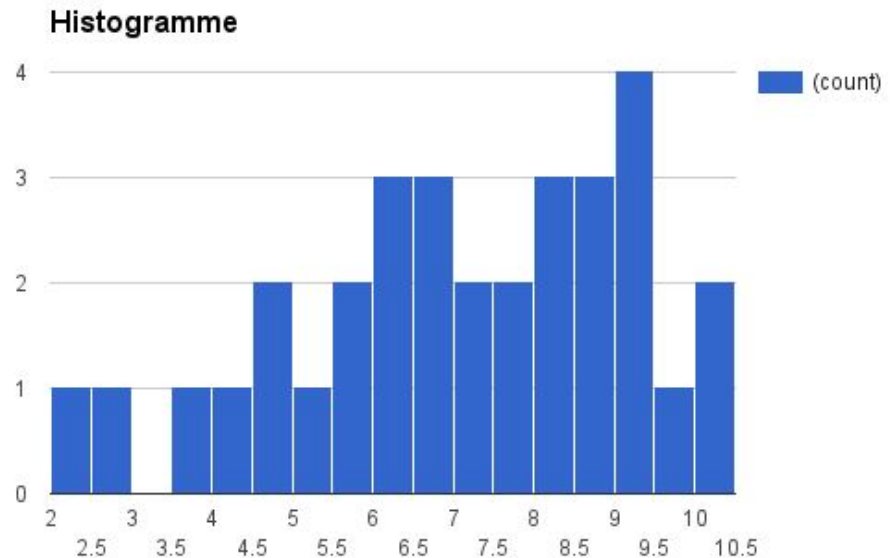
- Email and name properly set /0.5
- Commit messages quality /1
- Number of the question in commit messages /0,5
- **Commit with menu.txt /1**
- **Commit of log.txt /0,5**
- **Commit of diff.txt /0,5**
- **First Merge Conflict /1**
- Develop was pushed /1
- Develop was merged /1
- Tag v1.0 /1
- Develop was updated /0,5
- Git flow: feature /1
- Git flow: release (including tag v2.0) /1

Total: 10,5

# Some statistics

- 32 repositories
- Only 20 have added the question # in commit messages
- 22 have made the first merge completely properly
- 19 have tagged v1.0 with expected history
- 10 have started the git flow part.

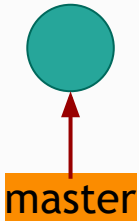
- Average mark: 6,72/10
- Median: 6,75
- Minimum: 2
- Maximum: 10,25



# III. Basic actions

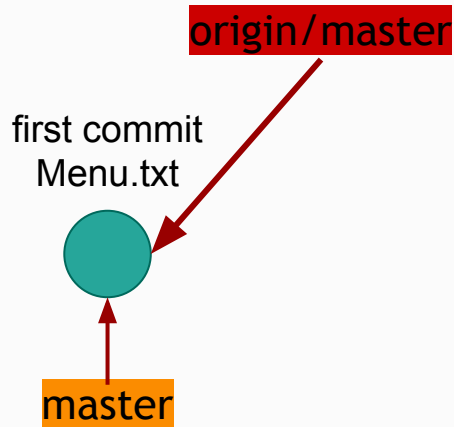
# Expected repository - III.4

first commit  
Menu.txt



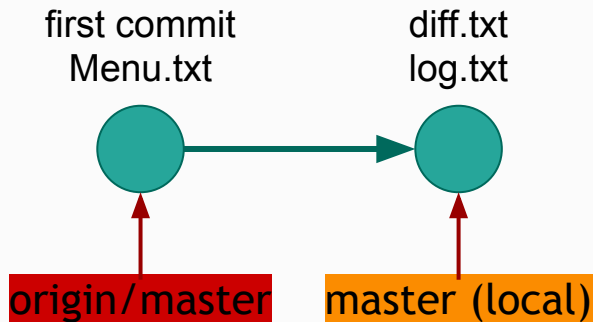
```
git add Menu.txt  
git commit -m "create menu."
```

# Expected repository - III.5



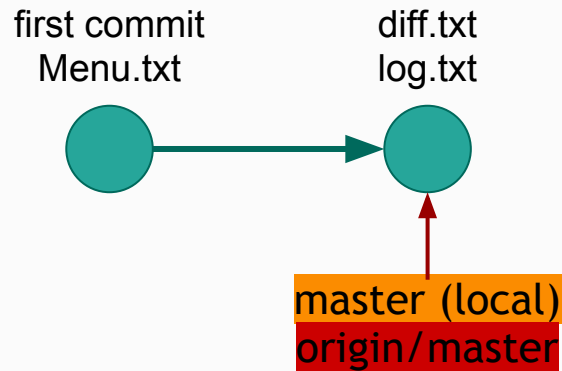
```
git push origin master
```

# Expected repository - III.8



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

# Expected repository - III.9

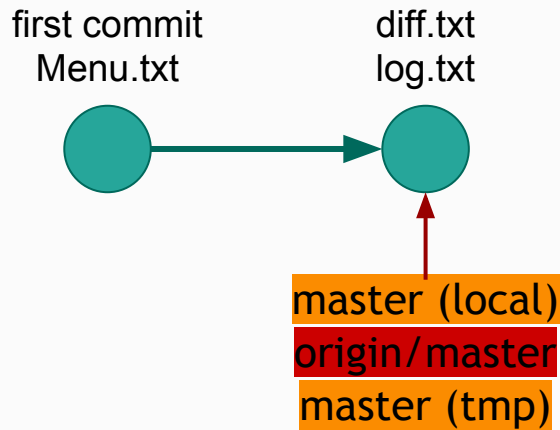


```
git push origin master
```



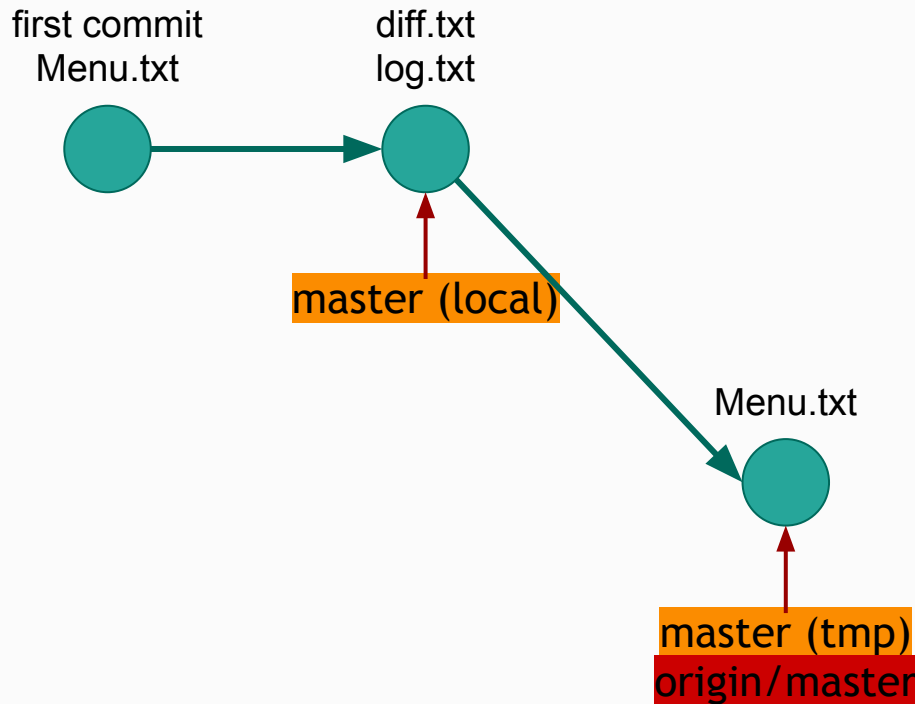
# IV. Conflicts

# Expected repository - IV.1



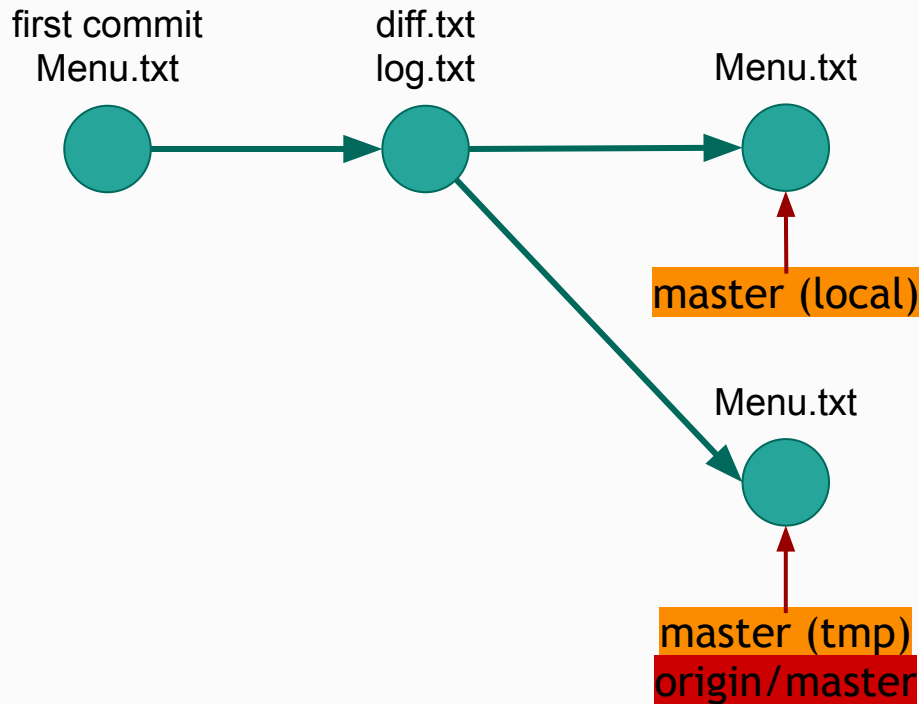
```
cd ../tmp  
git clone https://url.git
```

# Expected repository - IV.2



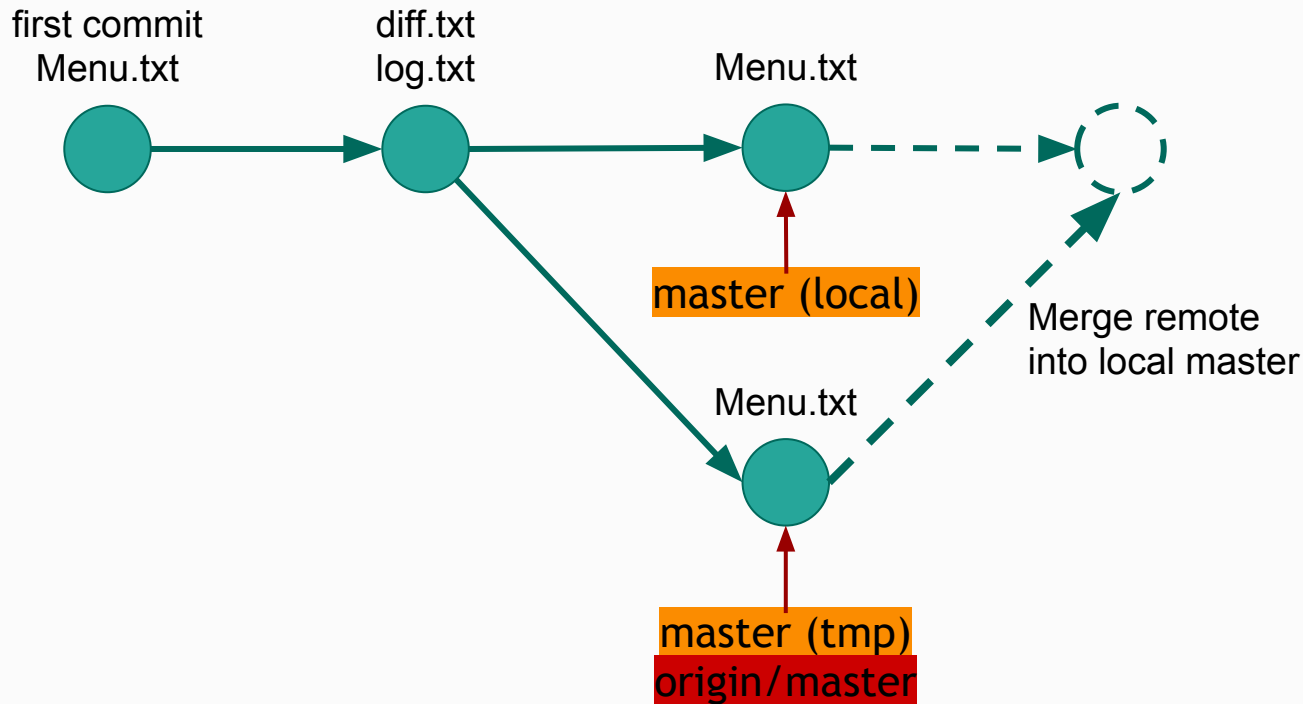
```
git add Menu.txt
git commit -m "modif menu in tmp folder"
git push origin master
```

# Expected repository - IV.4



```
cd previous_repo
git add Menu.txt
git commit -m "modif menu in base folder"
```

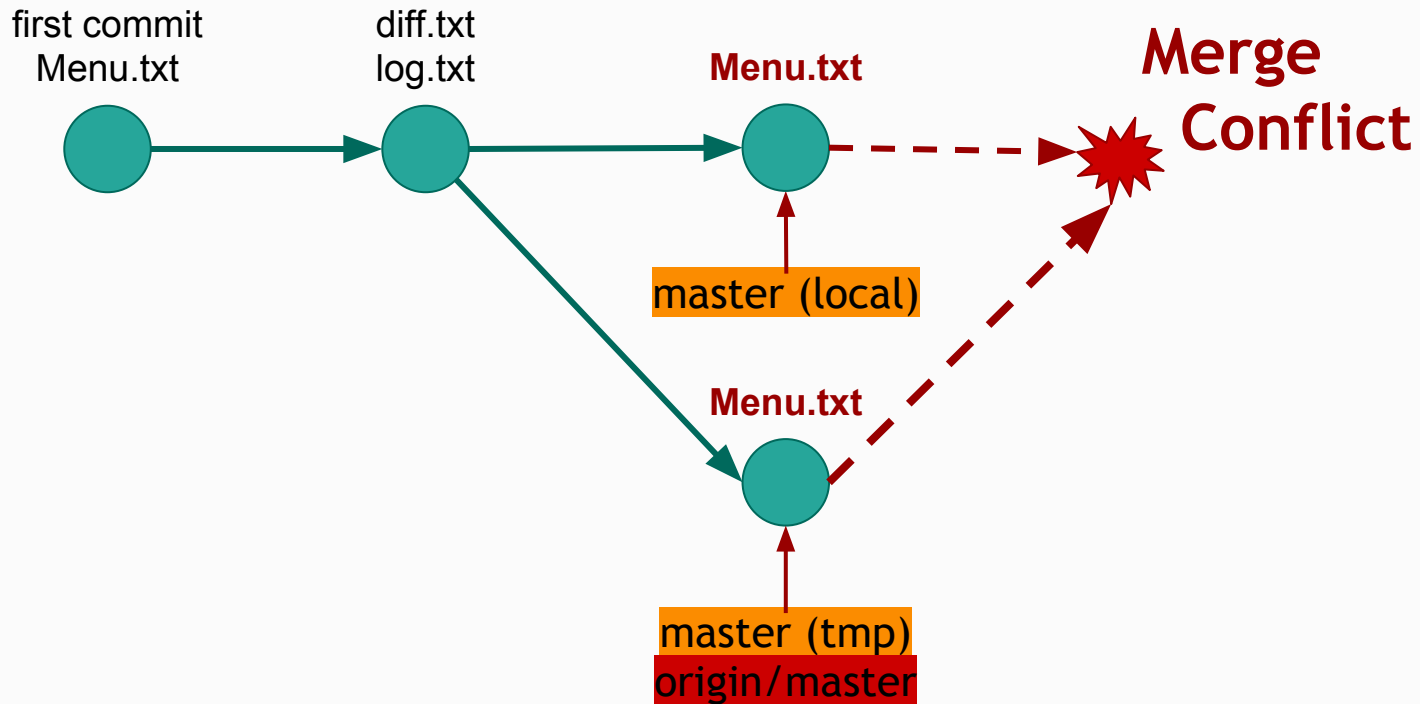
# Expected repository - IV.5



```
git push origin master  
git pull origin master
```

**does not work**

# Expected repository - IV.5

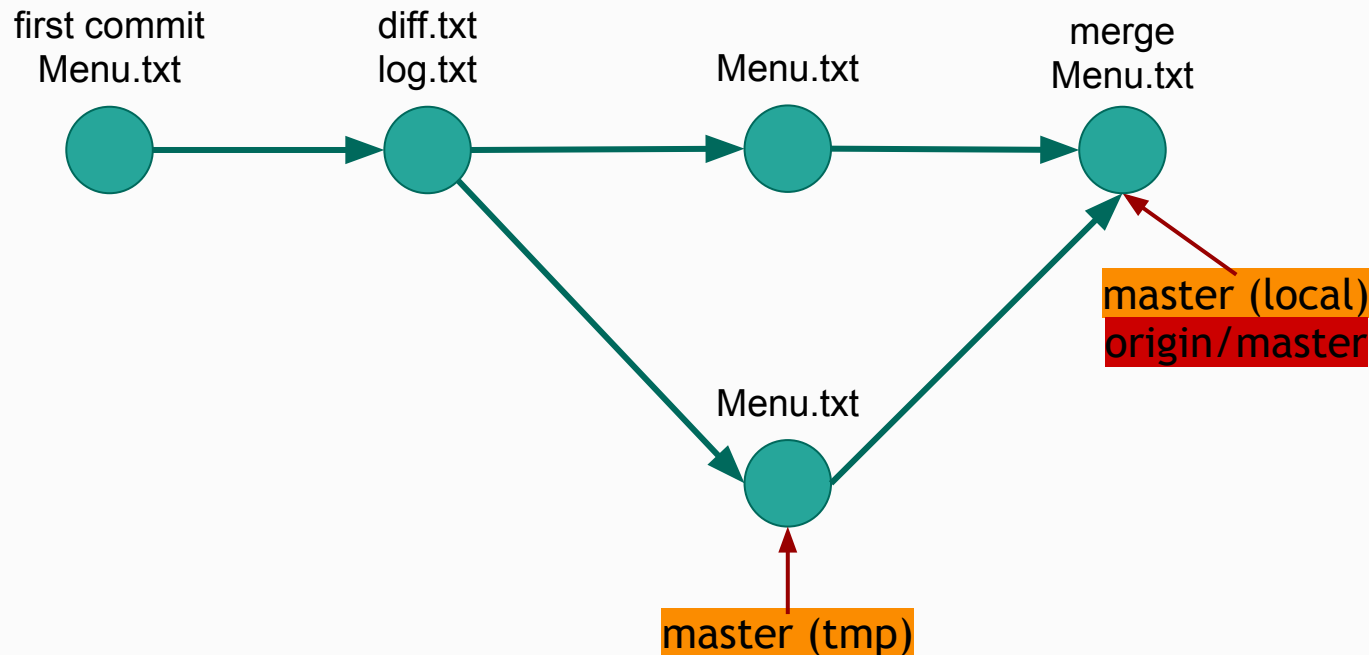


**\* resolve conflicts \***

```
git add Menu.txt
```

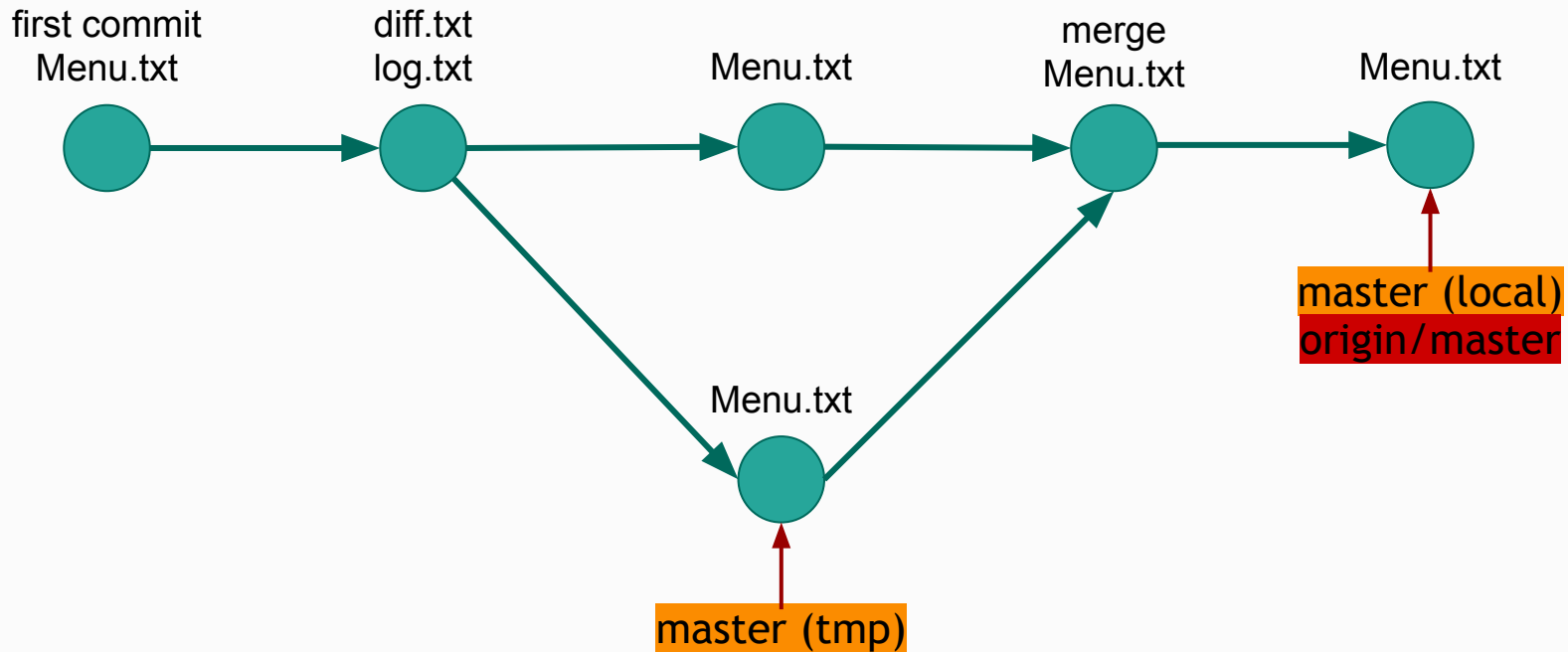
```
git commit -m "merge origin/master into master"
```

# Expected repository - IV.8



```
git push origin master
```

# Expected repository

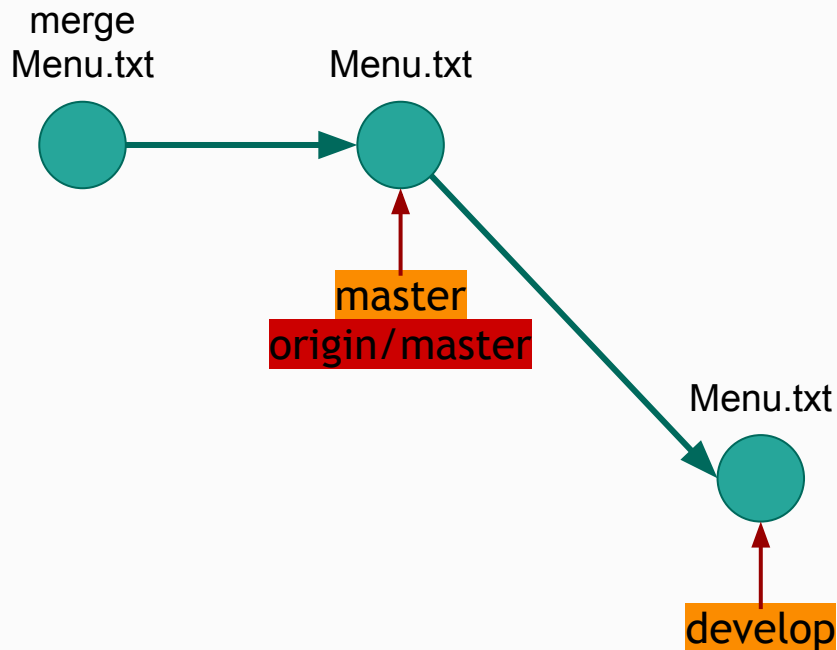


**more commits**



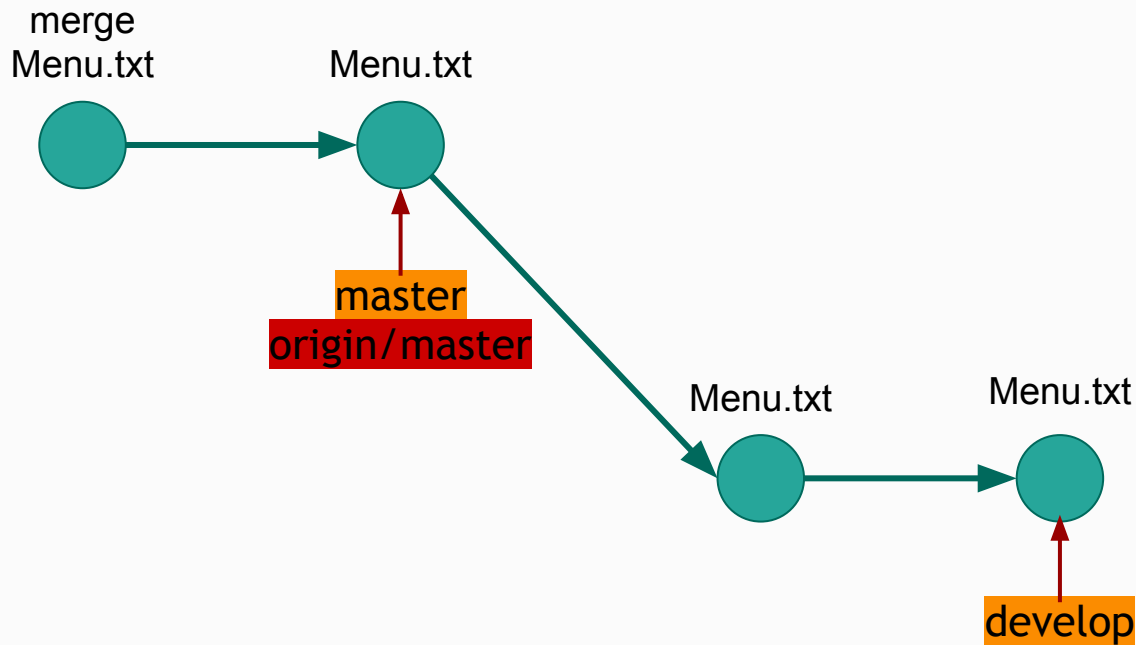
# V. Branches

# Expected repository - V.3



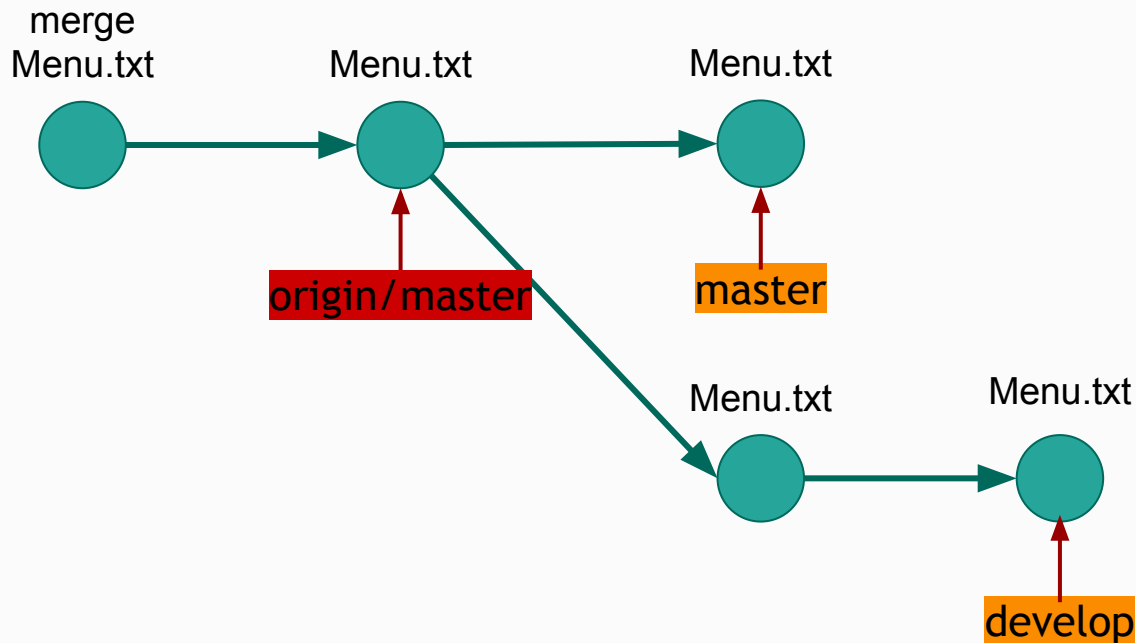
```
git checkout -b develop
git add Menu.txt
git commit -m "modif menu on develop"
```

# Expected repository - V.4



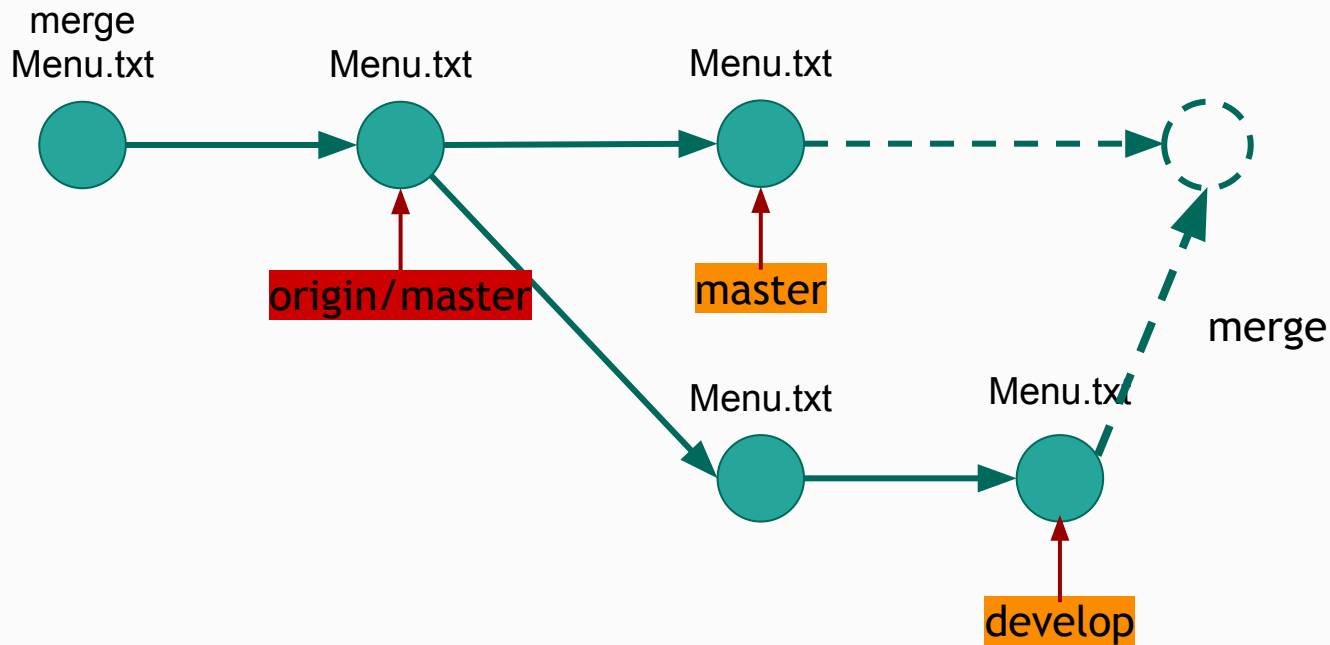
```
git add Menu.txt  
git commit -m "modif menu on develop"
```

# Expected repository - V.6



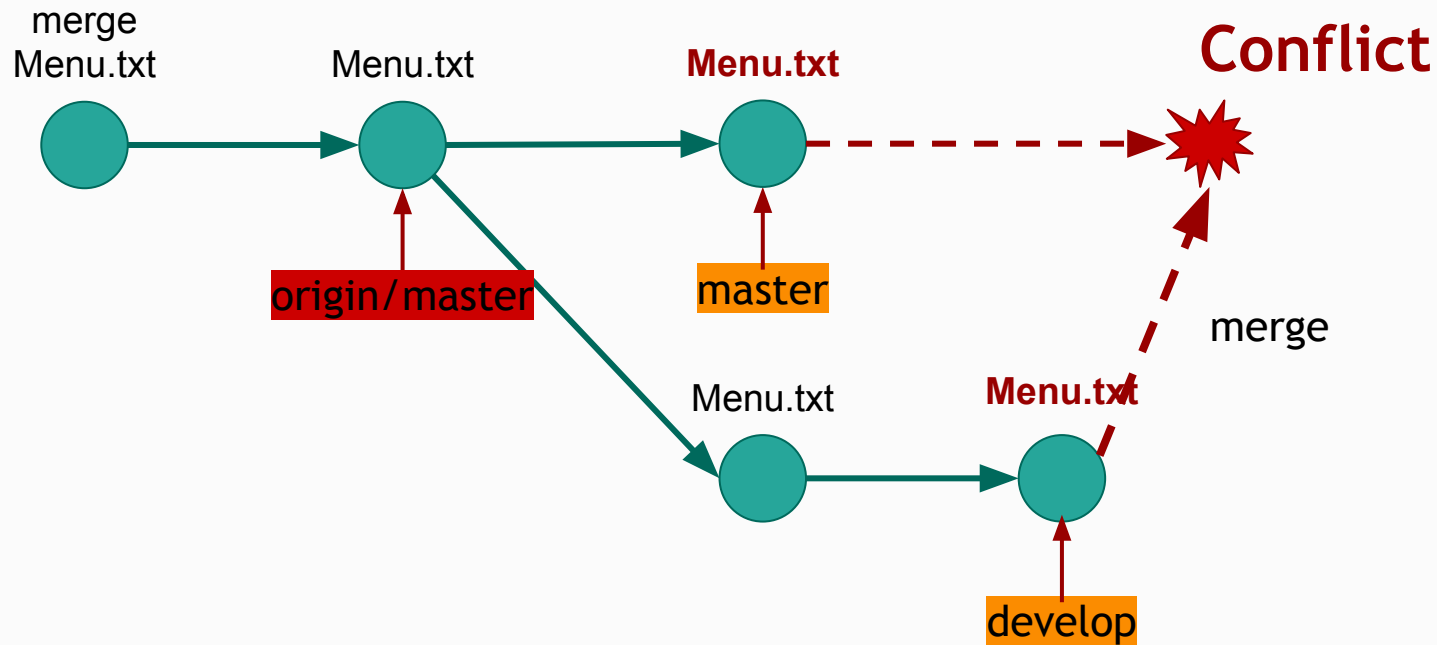
```
git checkout master
git add Menu.txt
git commit -m "modif menu on master"
```

# Expected repository - V.7

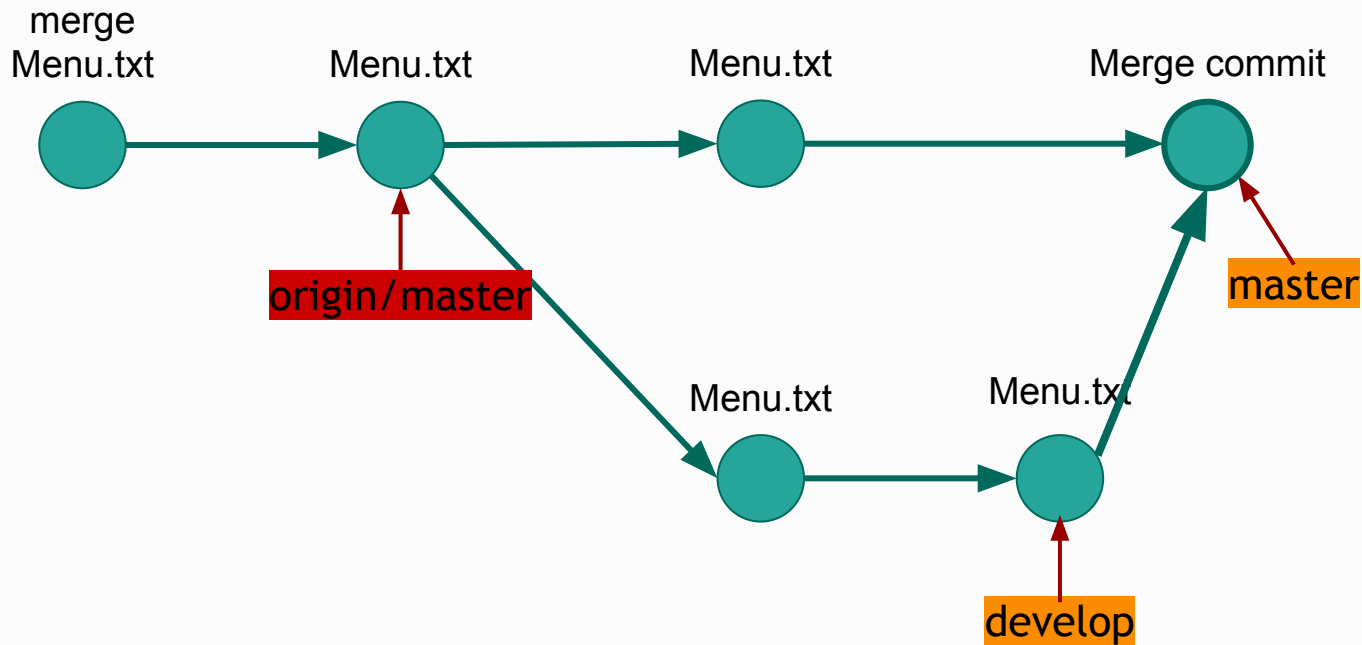


```
git merge develop
```

# Expected repository - V.7

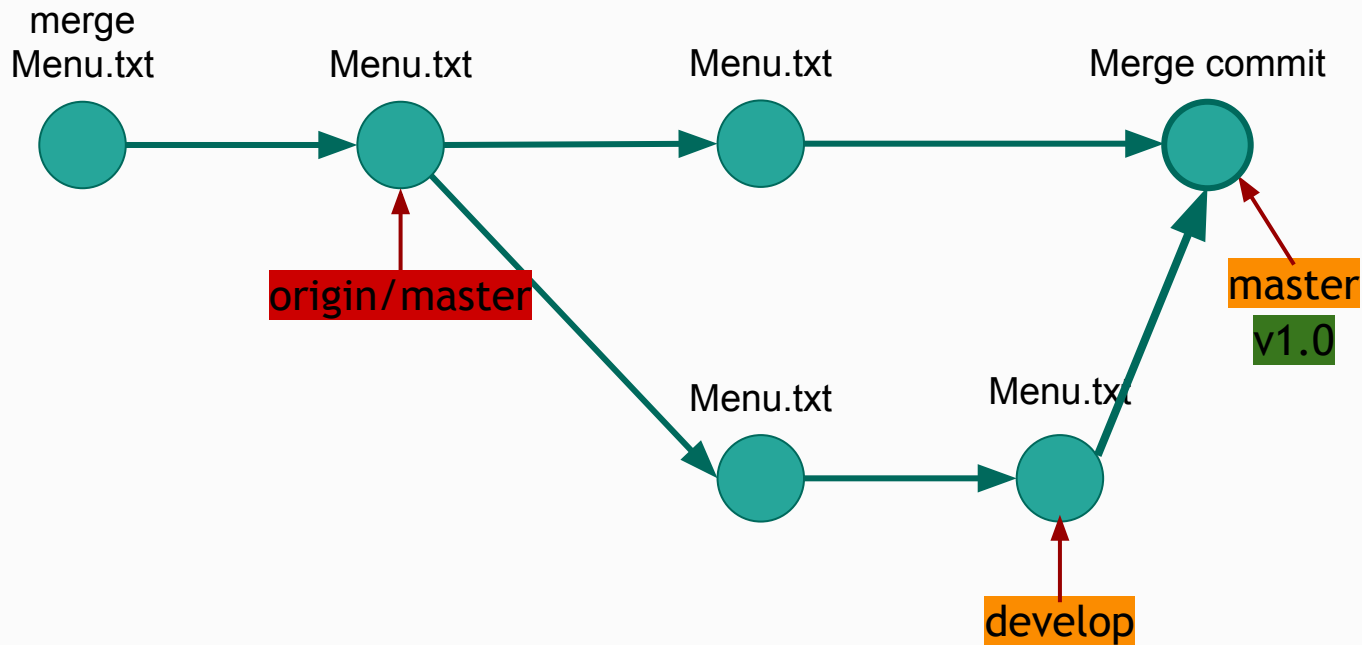


# Expected repository - V.8



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

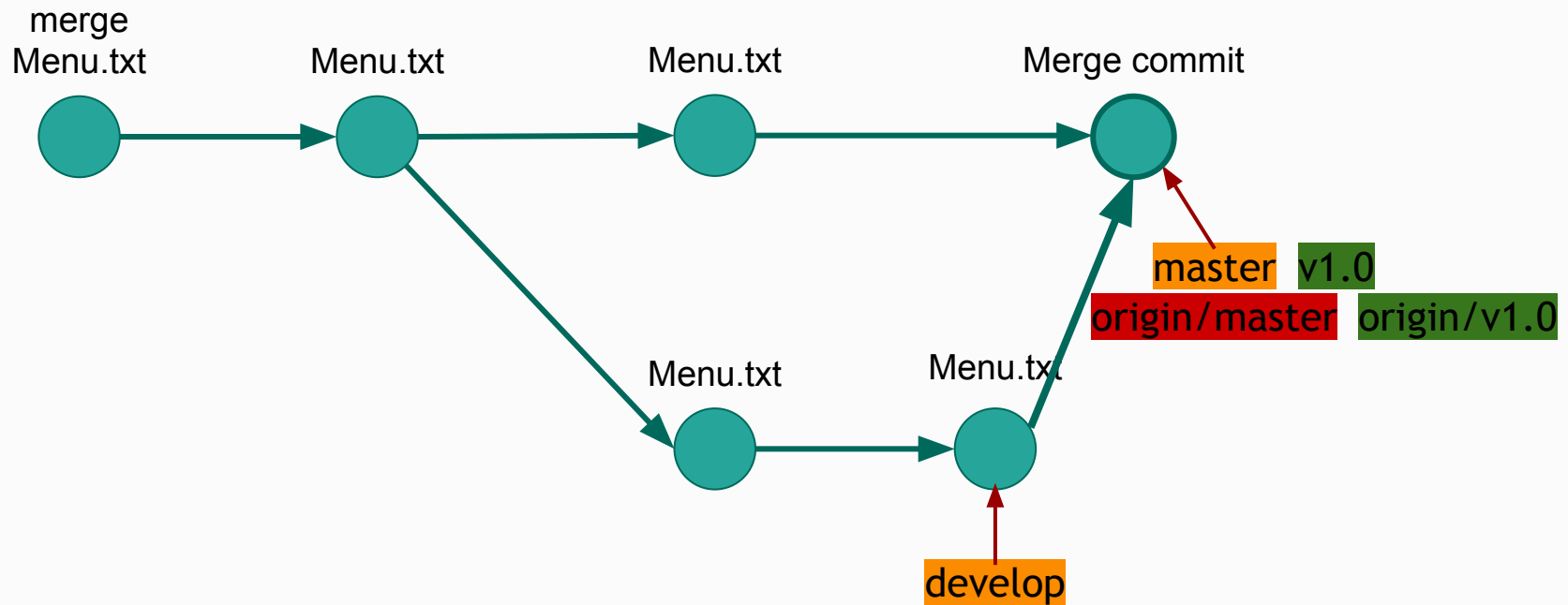
# Expected repository - V.10



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

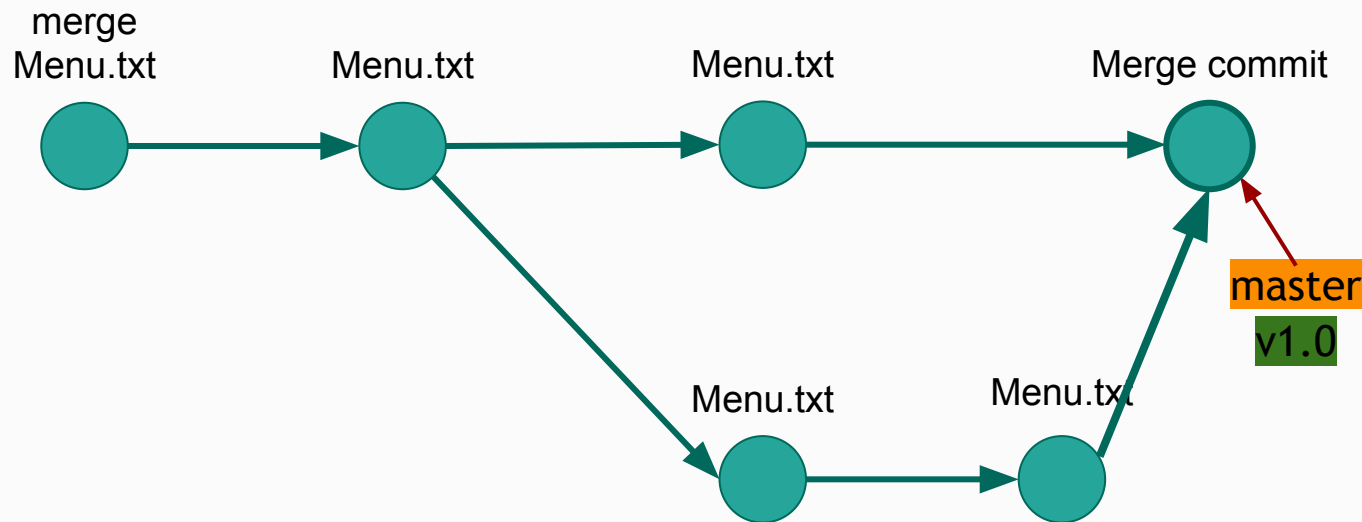


# Expected repository - V.12



```
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"

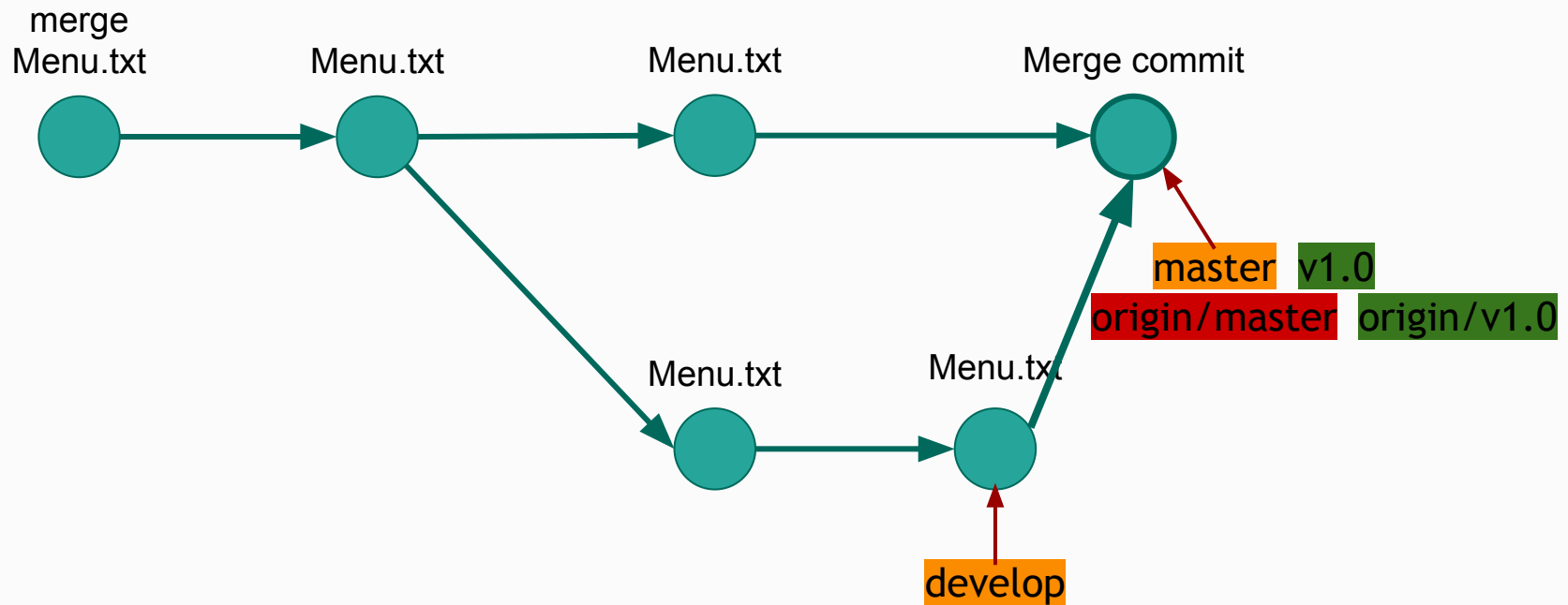
master v1.0

committed 8 days ago

2 parents c63f63a + 3030a56

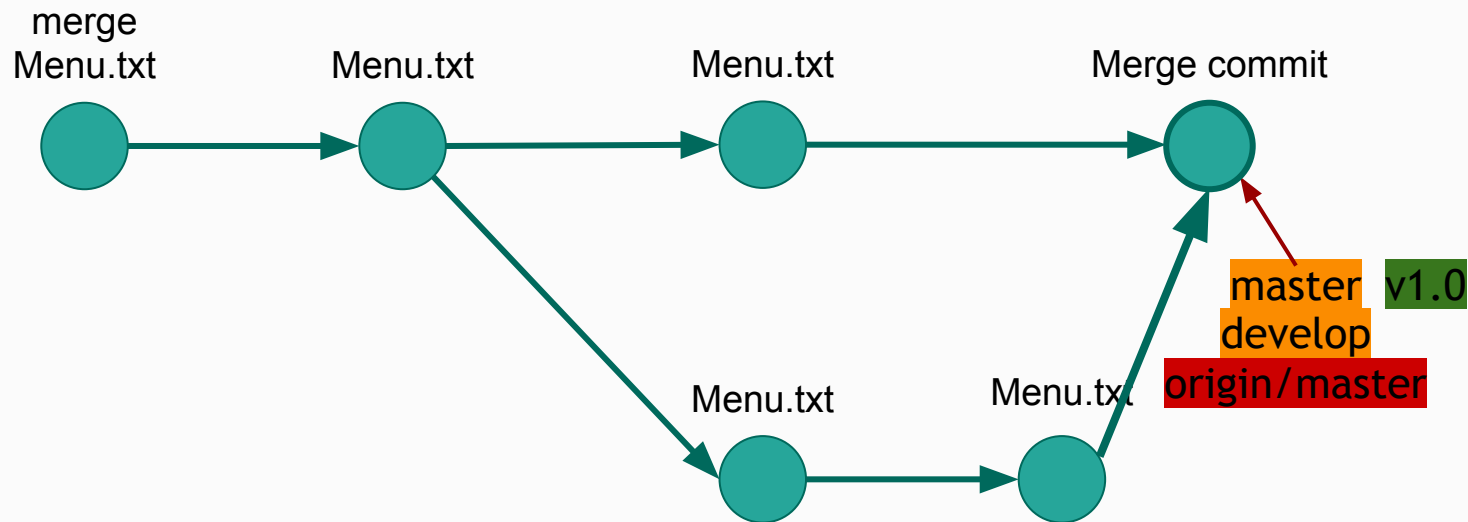
commit 5f104a6789fe4bfb

# Expected repository - V.12



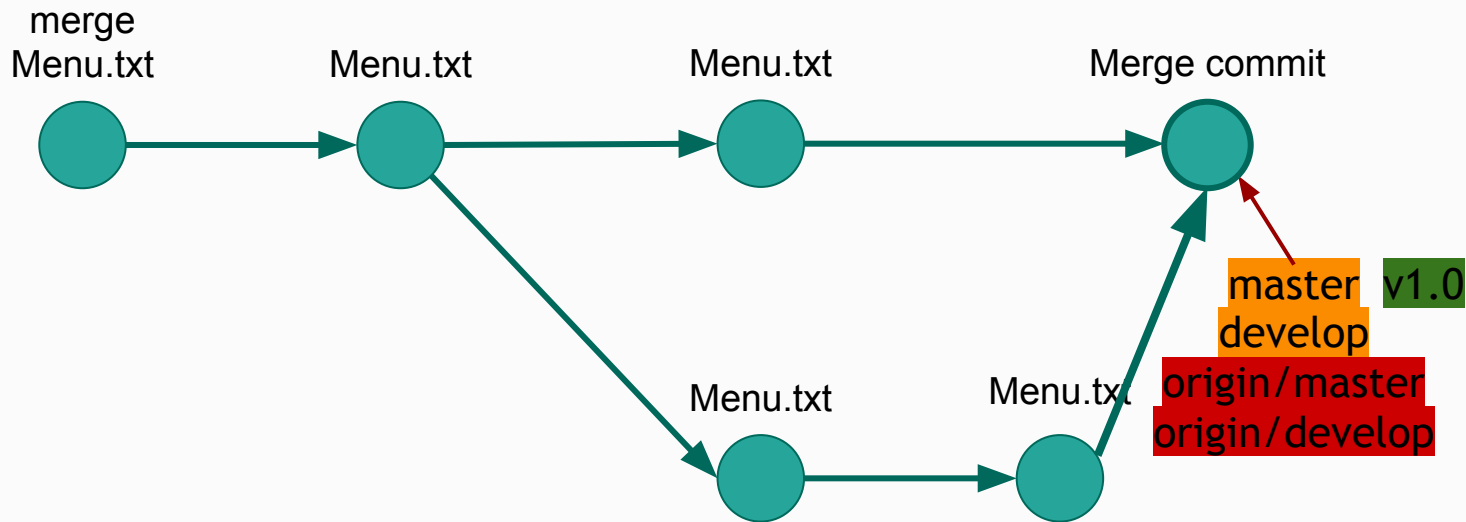
```
git push origin master  
git push origin v1.0
```

# Expected repository - V.14



```
git checkout develop
git merge master
```

# Expected repository - V.15



```
git push  
git push origin develop
```

does not work

# VI. Git Flow

# Git flow feature start

```
git flow feature start platLasagnes
```



```
git checkout develop
```

```
git branch feature/platLasagnes
```

```
git checkout feature/platLasagnes
```

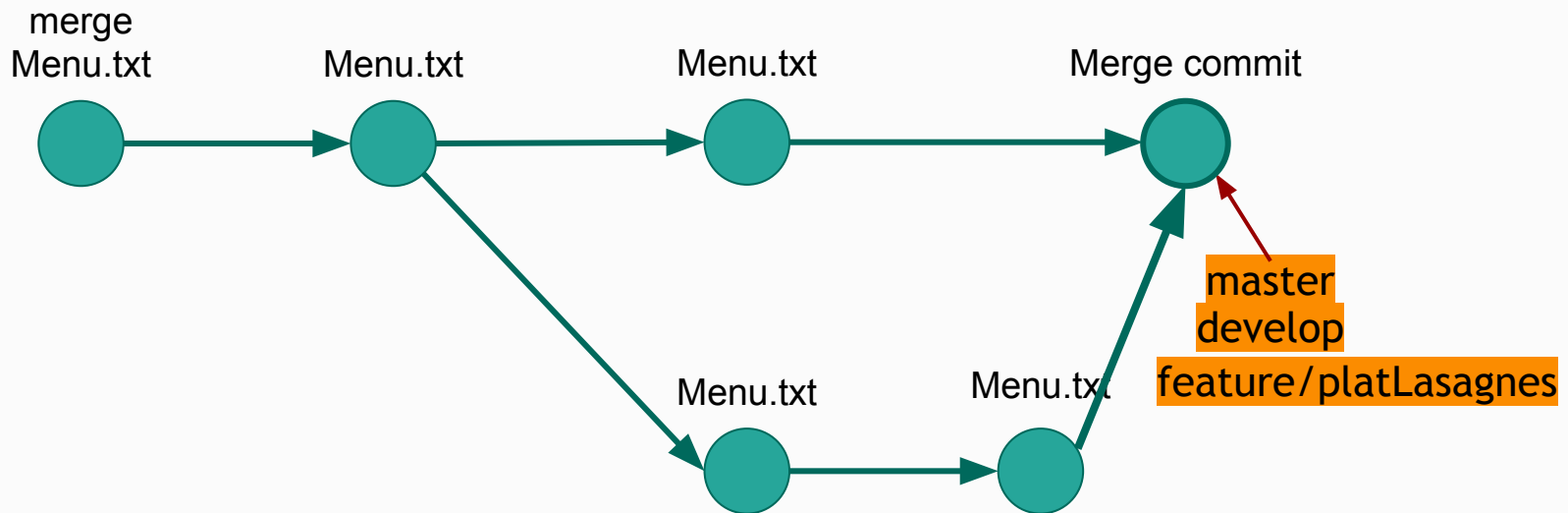
```
$ git flow feature start platLasagnes
Switched to a new branch 'feature/platLasagnes'

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
```

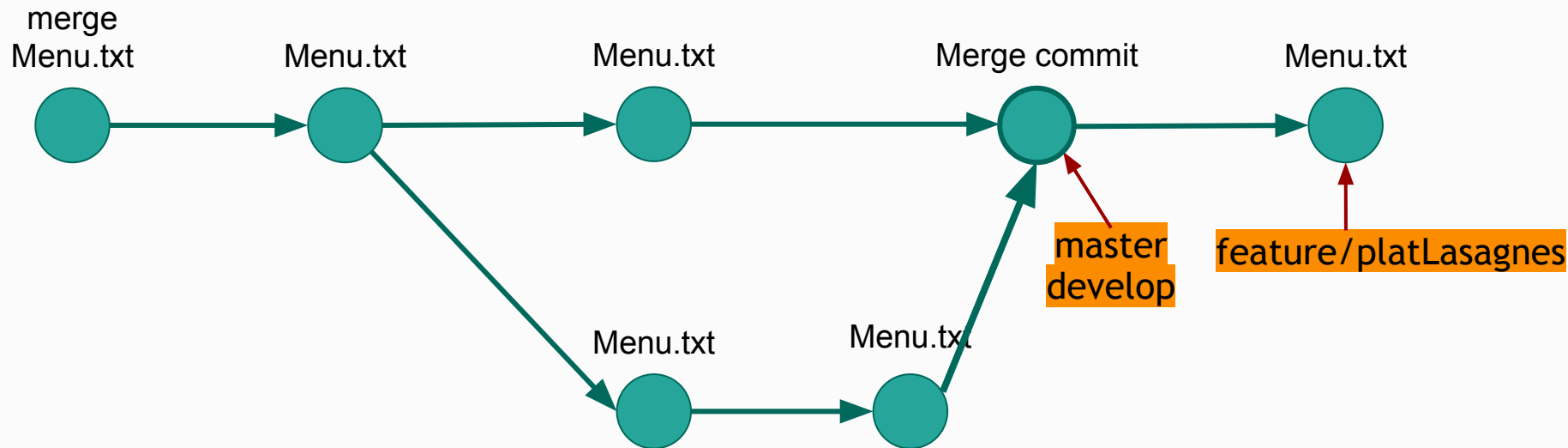
# Expected repository - V1.2



- Features start from **develop**



# Expected repository - V.5



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

# Git flow feature finish

```
git flow feature finish platLasagnes
```



```
git checkout develop
```

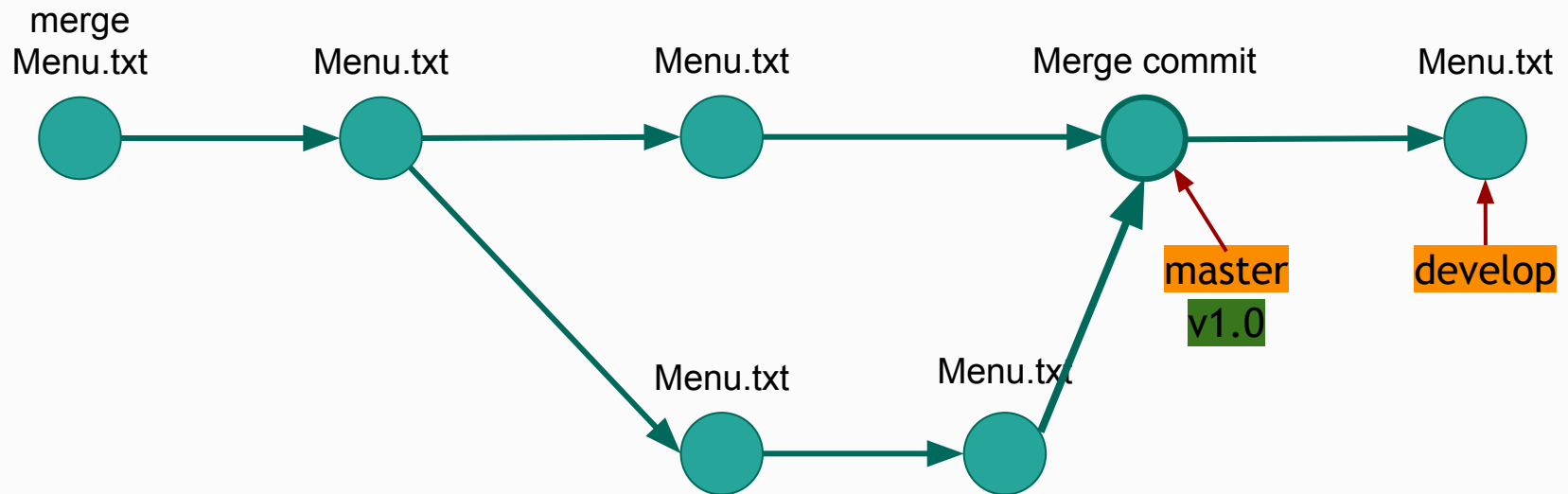
```
git merge feature/platLasagnes
```

```
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).

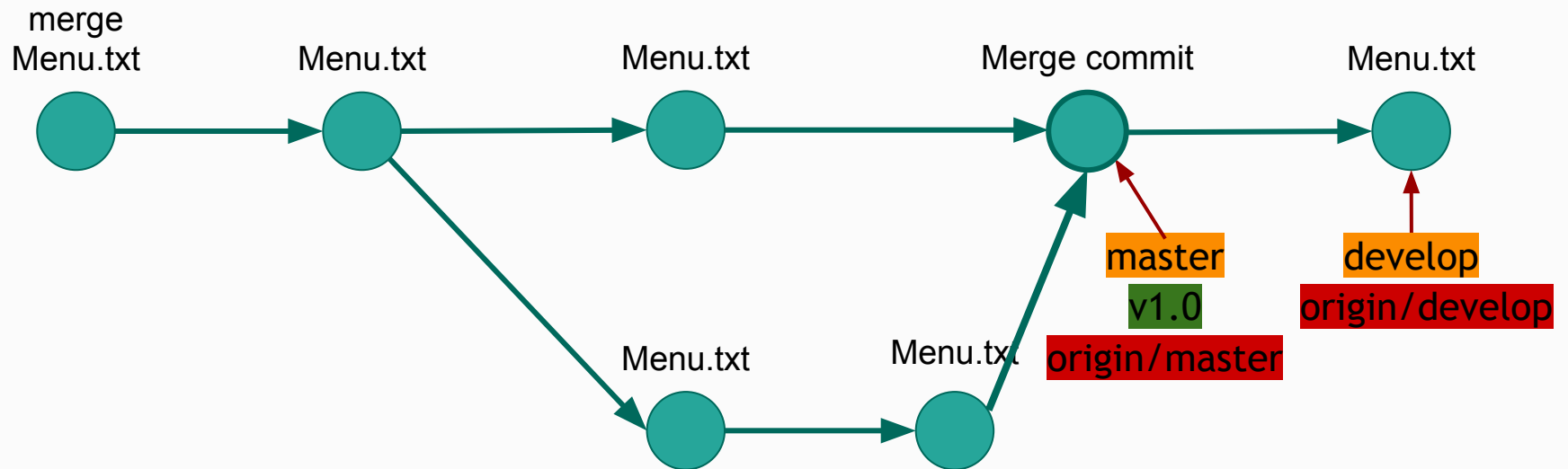
Summary of actions:
- The feature branch 'feature/platLasagnes' was merged into 'develop'
- Feature branch 'feature/platLasagnes' has been locally deleted
- You are now on branch 'develop'
```

# Expected repository - V1.6



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

# Expected repository - V1.8



```
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'

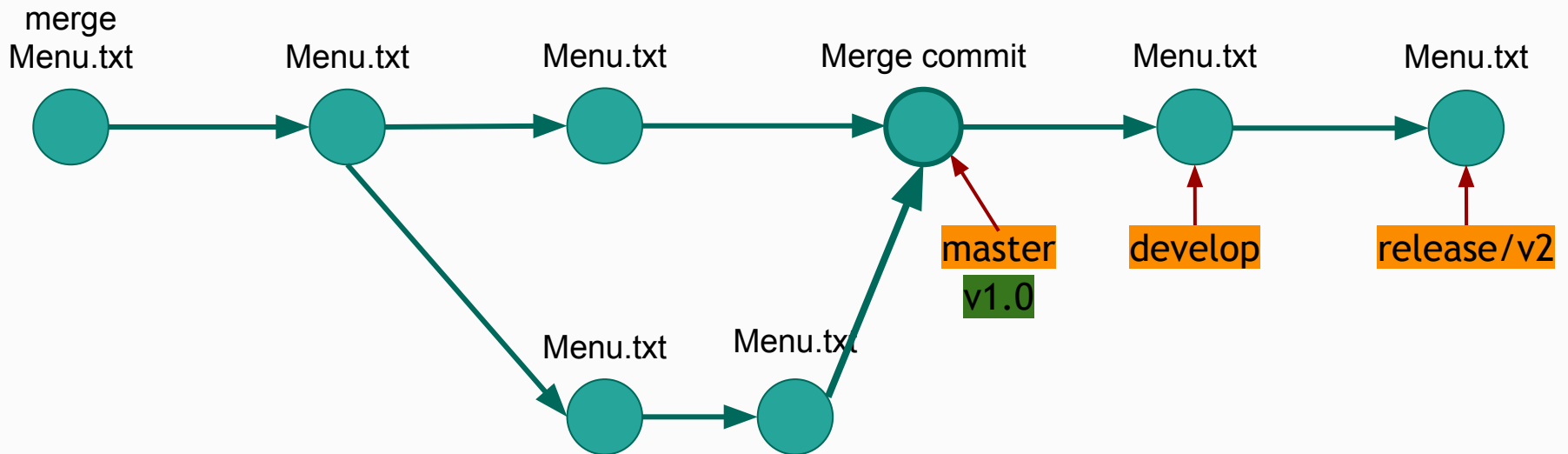
Follow-up actions:
- Bump the version number now!
- Start committing last-minute fixes in preparing your release
- When done, run:

    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).

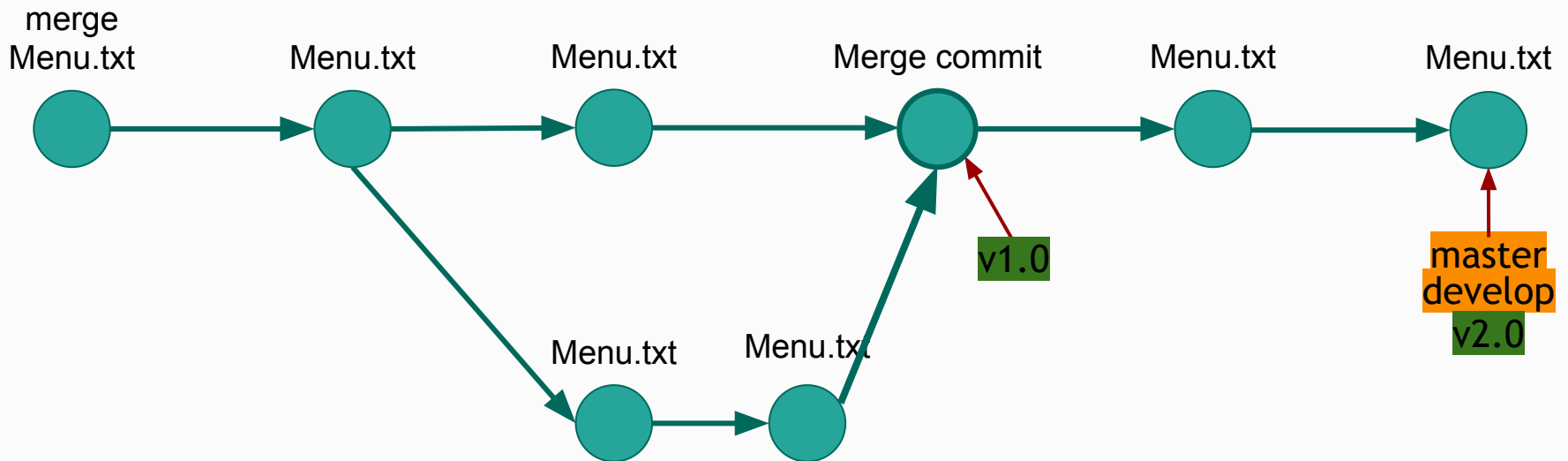
Summary of actions:
- Release branch 'release/v2' has been merged into 'master'
- The release was tagged 'v2'
- Release tag 'v2' has been back-merged into 'develop'
- Release branch 'release/v2' has been locally deleted
- You are now on branch 'develop'
```

# Expected repository - V1.10



- Releases start from **develop**

# Expected repository - V1.11



- Merge in master and develop
- Tag created automatically
- Branche release/v2 was deleted

# Summary



# Common mistakes

- Manual merges not done properly
  - > Don't forget to add the files before committing
  - > Make sure the file is correct before committing
- Diff file empty
  - > Make sure the file is correct before committing
- Unclear commit messages / No question number
  - > Hard to know where you stopped.
- Branches or tags not pushed
  - > Even harder to know where you stopped.

# 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).

Sprint-0

# Some more statistics

- 6 repositories
- None met all requirements
- 1 group did not create a release on Github -> < 5/10
- 1 person did not join their group on Github -> 0
- Only 3 created the release on Jira
- Only 2 have properly defined a service
- No repository was completely clean (1 is 72MB!!)
- Some releases were made ahead of time -> /2

# Some other remarks

- We look at the repo at the time of the tag, not at the latest commit.
- Be careful about folder and file names
  - I should know what a folder contains without opening
  - We ask for specific file names and locations (key.txt)

# Files to exclude

- Hidden files (.DS\_Store, ...)
- IDE files (.idea/, \*.iml, ...)
- Build directories (target/, build/, ...)
- Lib directories (.gradle/, ...)

Before committing a file, ask yourself if it is needed.

**If no, do not commit it.**



Time to wake up!

# Conclusion



