

# Git и gitflow

Что такое VCS?

Основные принципы Git

Удачная модель ветвления: gitflow

Работа с менеджерами репозиториев

# Системы контроля версий

Хранение версий документа

Хранение изменений

Git, SVN, Mercurial

# Системы контроля версий

## Sunsetting Mercurial support in Bitbucket



Denise Chan

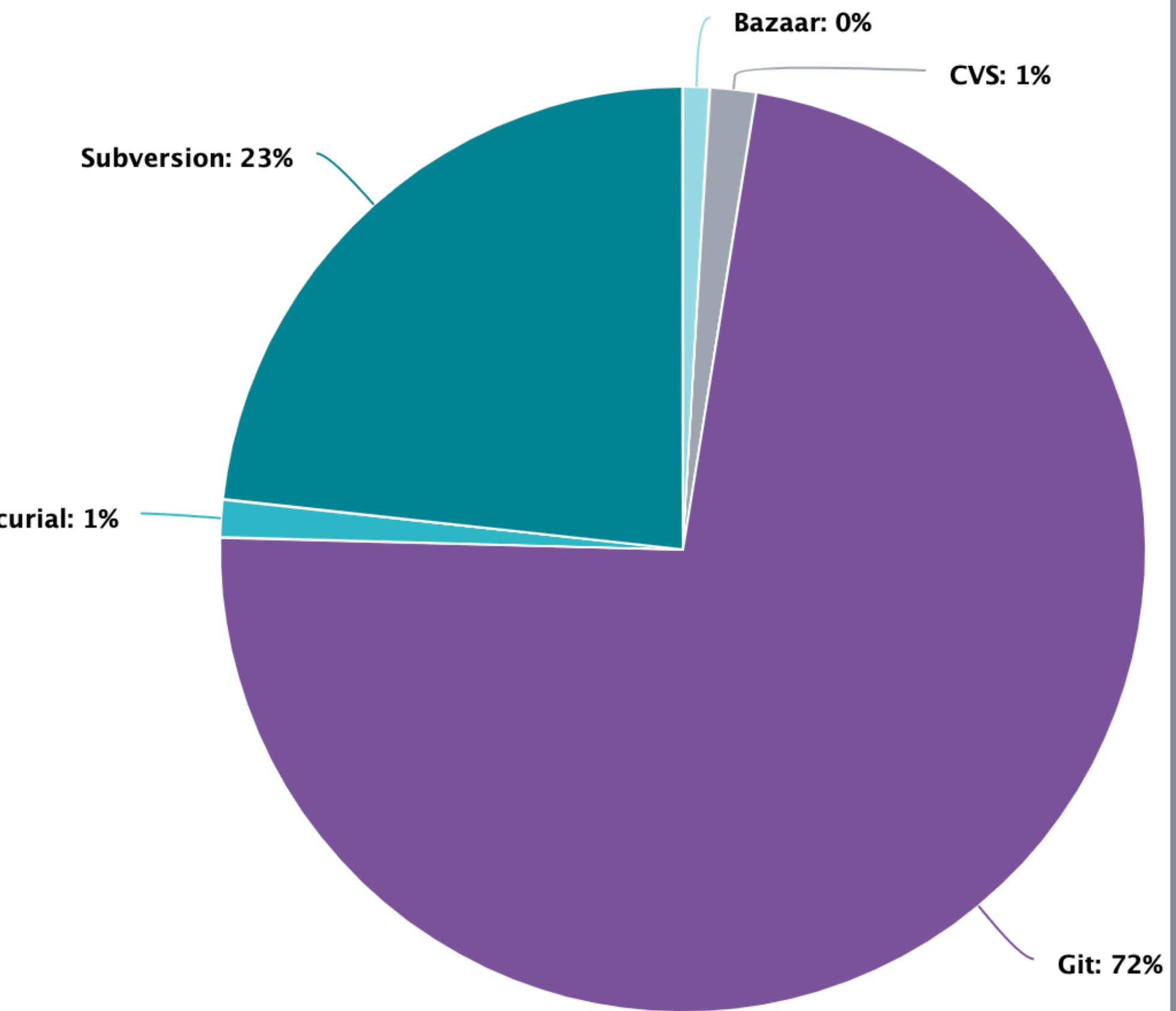
August 20, 2019 | 2 min read

The version control software market has evolved a lot since Bitbucket began in 2008. When we launched, centralized version control was the norm and we only supported Mercurial repos.

But Git adoption has grown over the years to become the default system, helping teams of all sizes work faster as they become more distributed.

As we surpass **10 million registered users** on the platform, we're at a point in our growth where we are conducting a deeper evaluation of the market and how we can best support our users going forward.

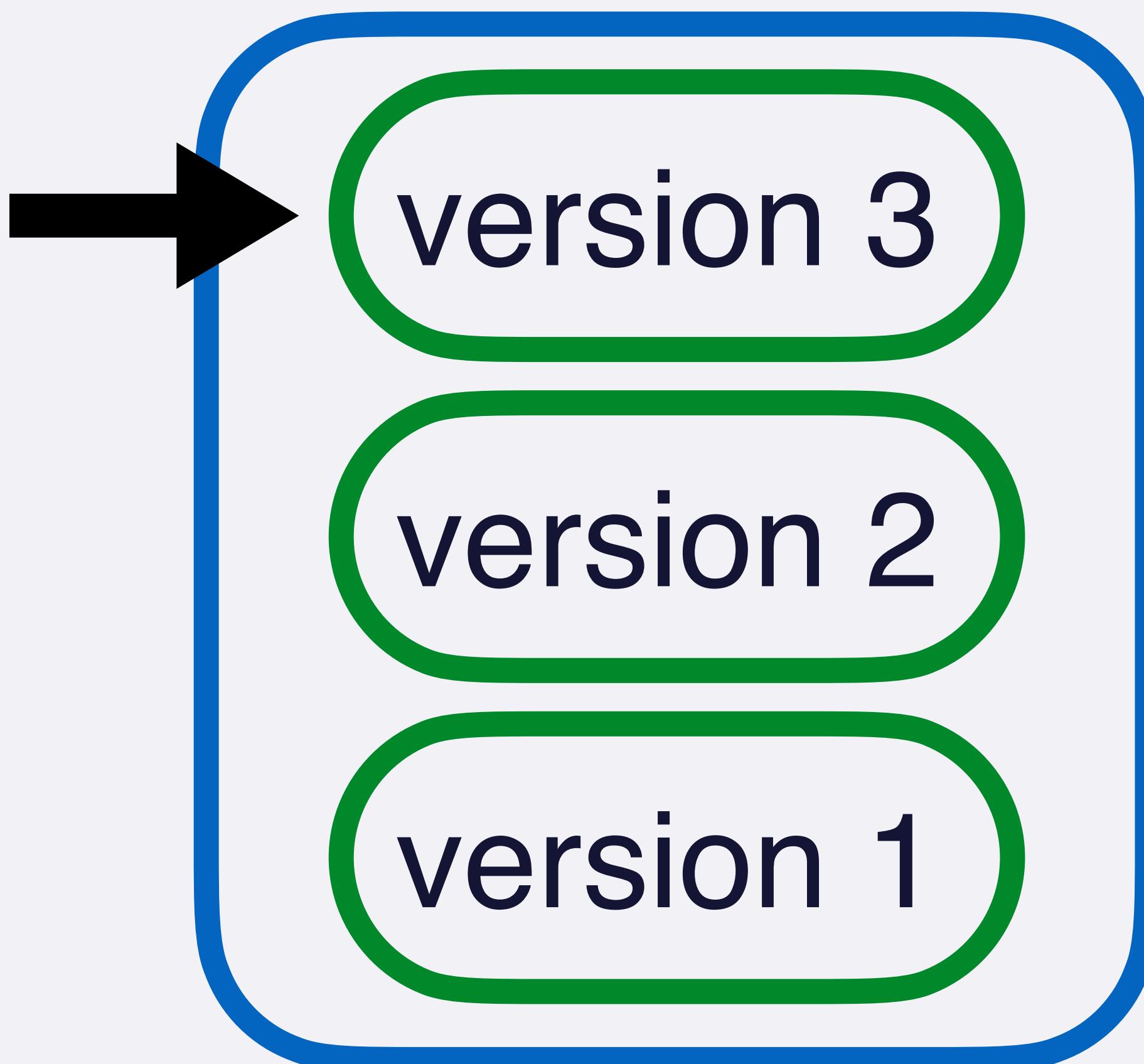
After much consideration, we've decided to remove Mercurial support from Bitbucket Cloud and its API. **Mercurial features and repositories will be officially removed from Bitbucket and its API on June 1, 2020.**



<https://bitbucket.org/blog/sunsetting-mercurial-support-in-bitbucket>

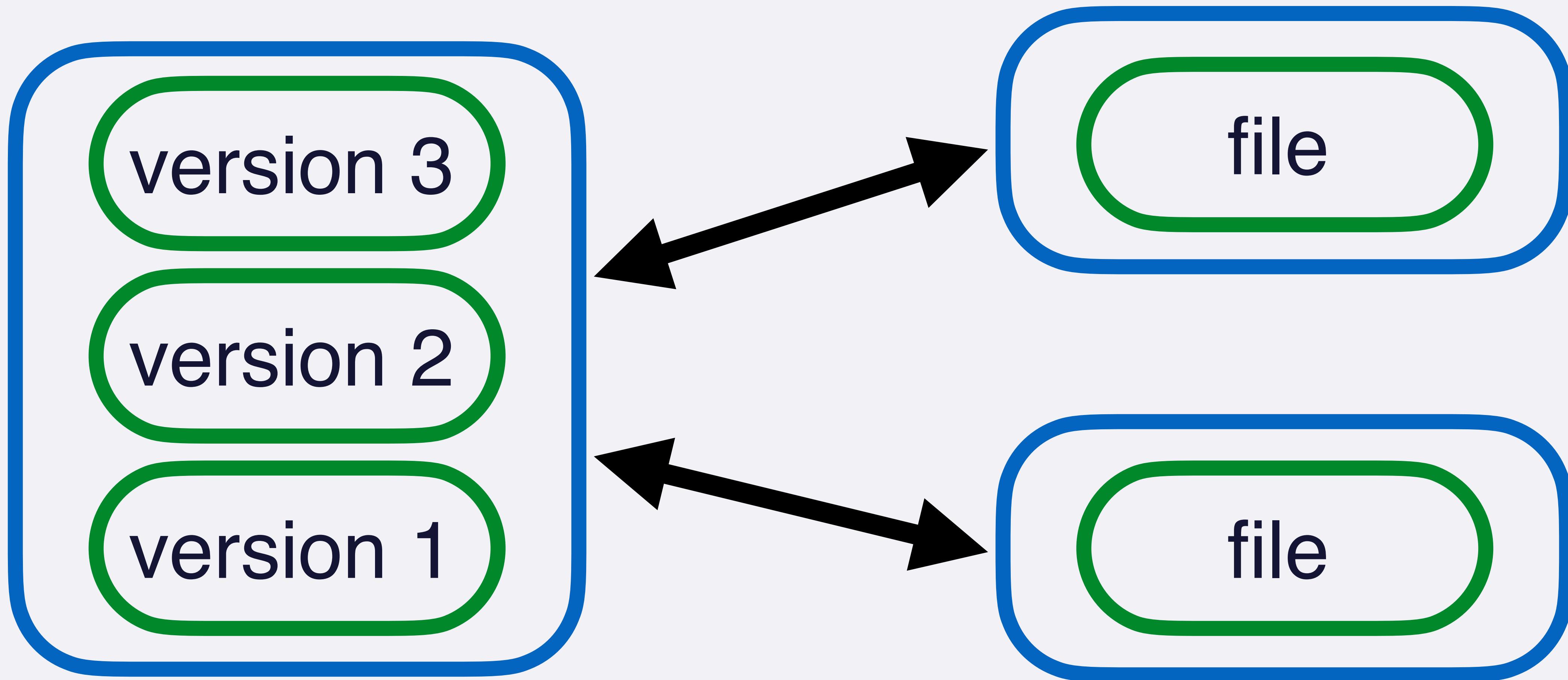
<https://www.openhub.net/repositories/compare>

# Локальные

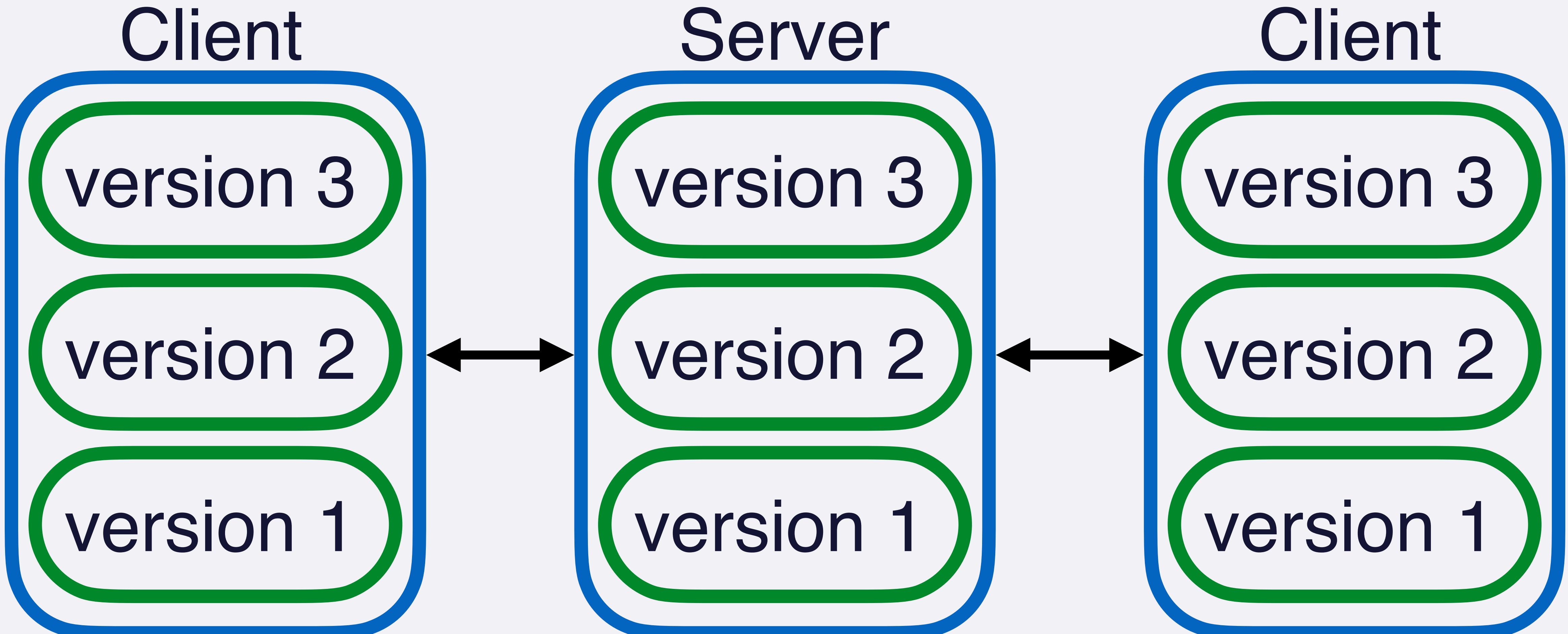


- Единственная копия
- Простота
- Небезопасность

# Централизованные



# Распределенные



# Git



Распределенная.  
Надежная.  
Твоя.

# Commit



Набор изменений

Атомарность

Свойства

- SHA-1
- автор
- дата
- сообщение

# Commit. Identifier



## SHA1 or SHA256

Not so fast, [Torvalds writes](#). Here's what he said:

**First off, the sky isn't falling. There's a big difference between using a cryptographic hash for things like security signing, and using one for generating a 'content identifier' for a content-addressable system like git.**

# Сообщения



**Начинать**



Илья Але... 362d46bf0cf  
Александр Але... 132689950b6  
Александр Але... РТРО-3023: исправлен дефект в продуктовой ан...  
конфига по п...  
правлена ошибка наи...

# Сообщения



## Причина изменений

Для устранения дефекта 12387 была исправлена утечка памяти

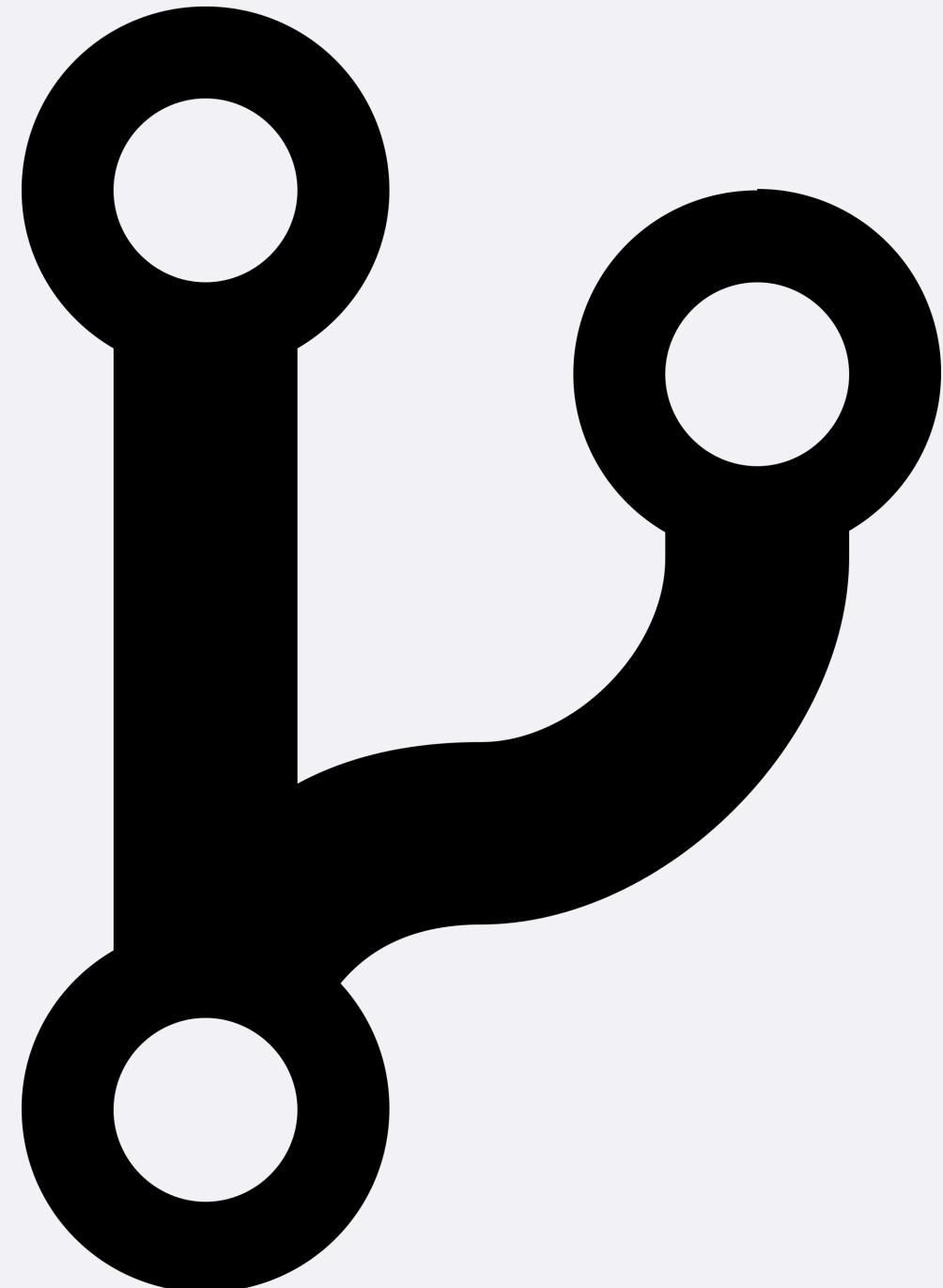
## Что сделано

- исправлена утечка памяти
- Написан модульный тест

## Ссылки

<ССЫЛКА НА СИСТЕМУ КРАШ-ЛОГОВ>

# Branch

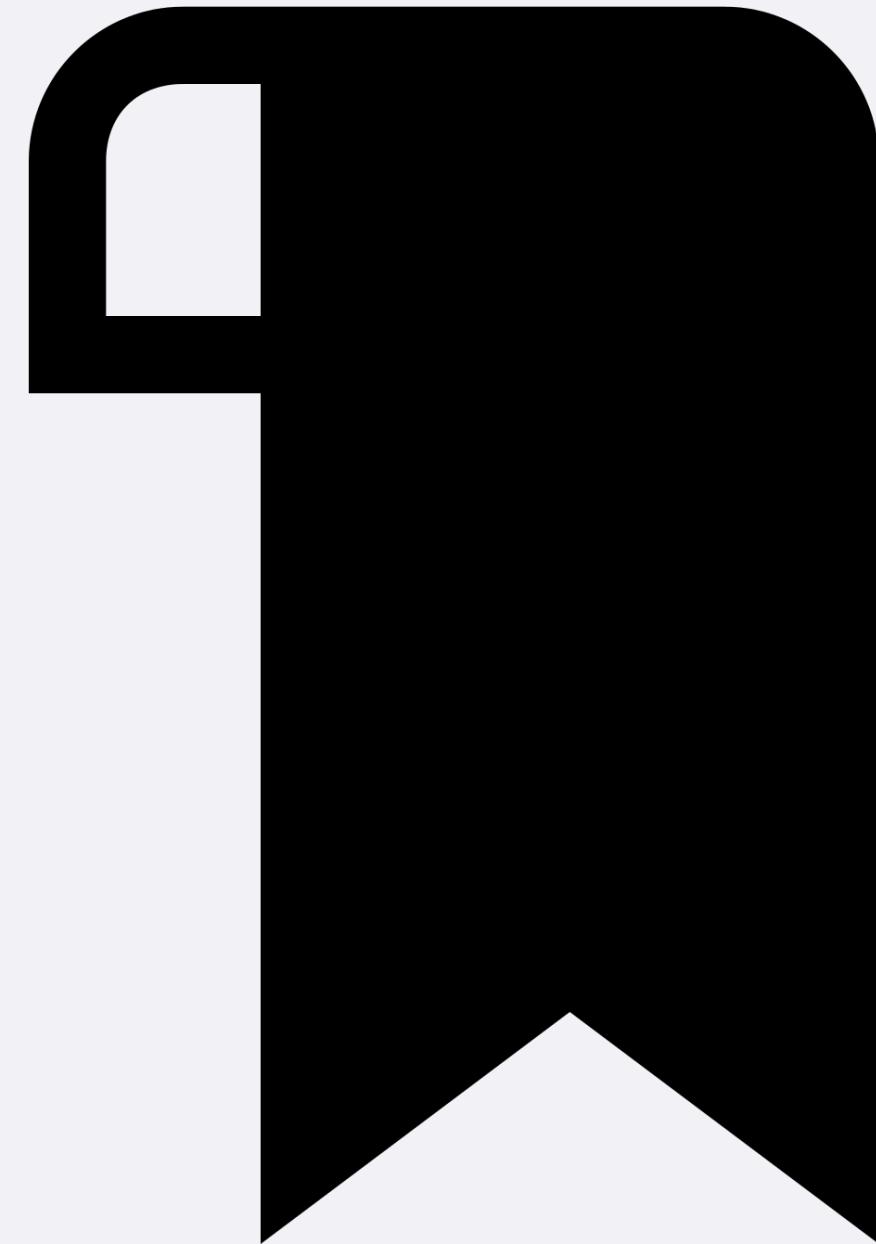


Последовательность  
коммитов

Имя

Указатель на последний

# Tag



Закладка

Имя

- develop/1.6.0/154
- release/1.5.4/102

# Инструменты

Терминал

SourceTree

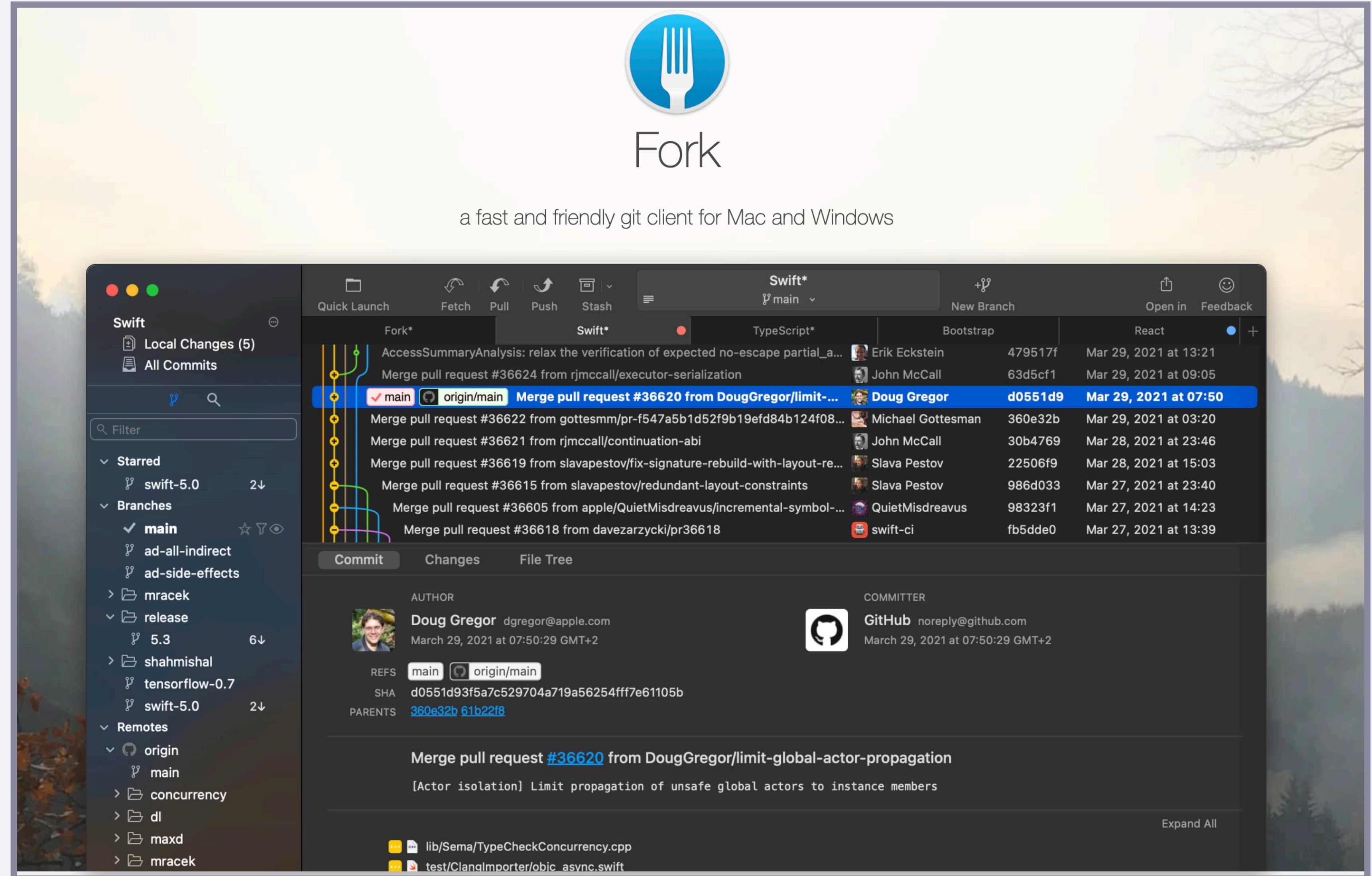
Fork

Tower

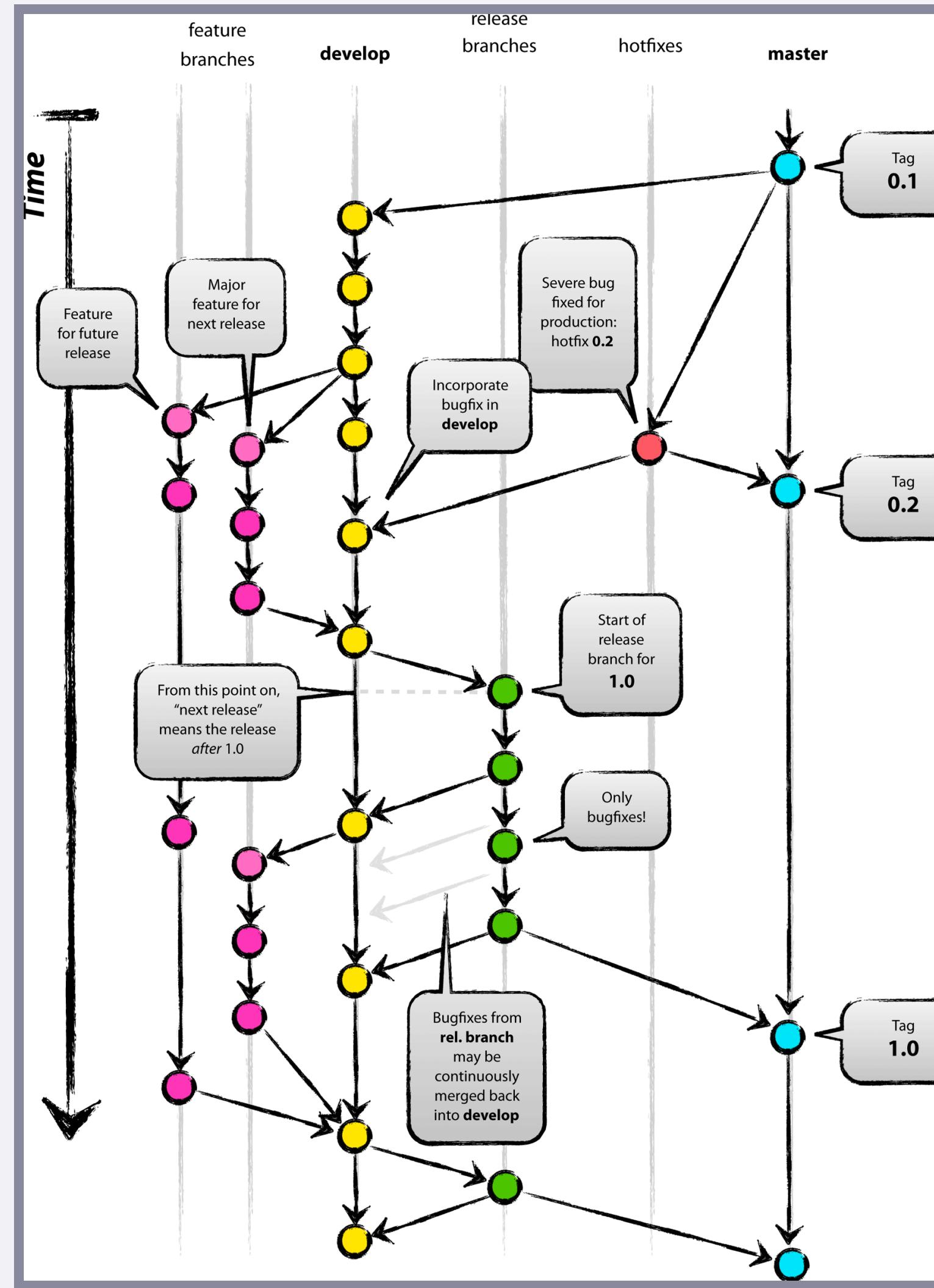


# Инструменты

Fork



# gitflow

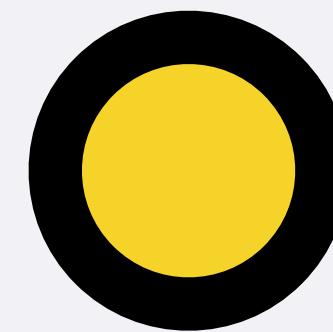
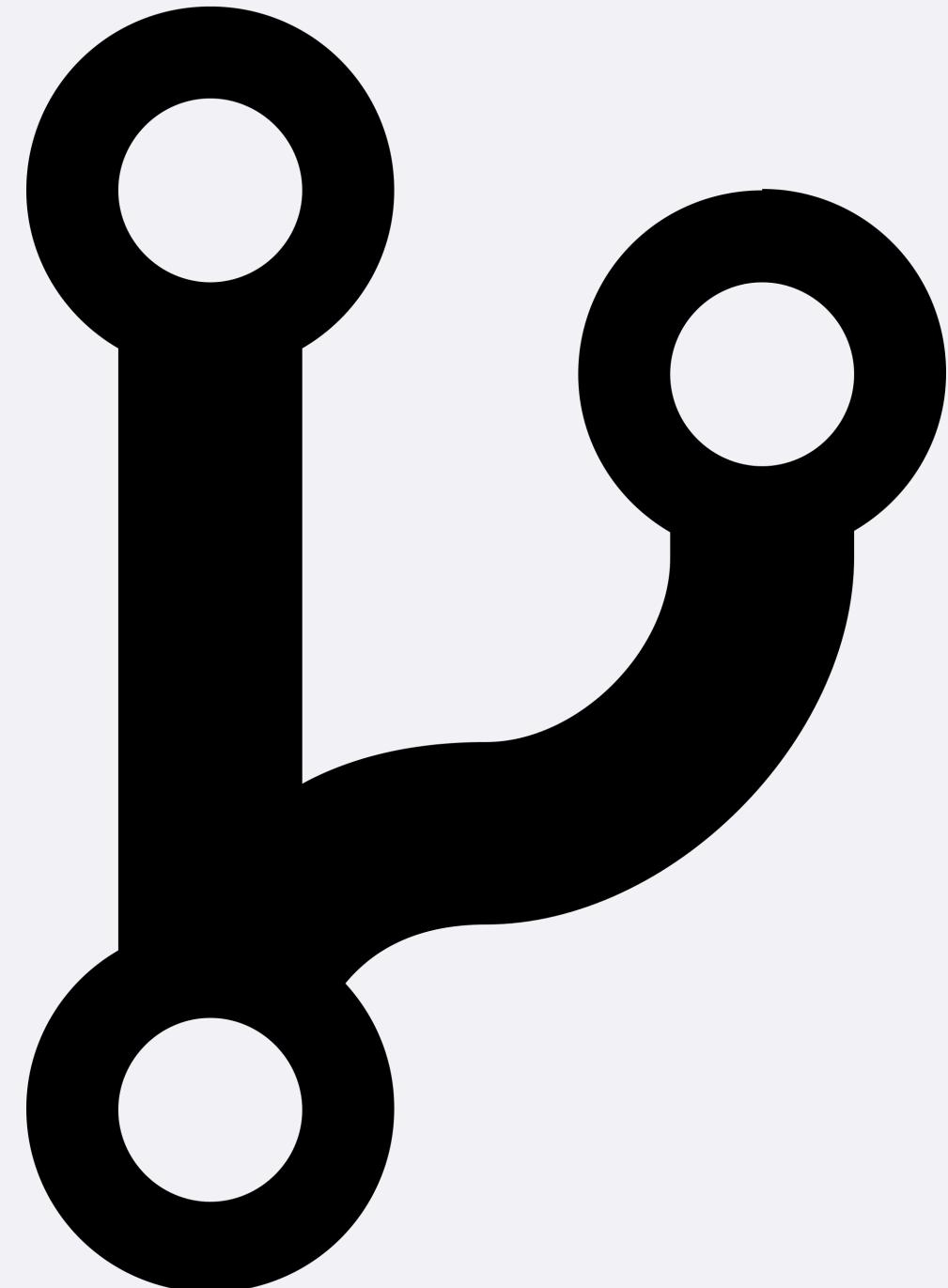


Роли веток

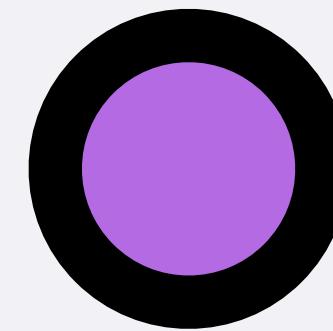
Удобство

Строгость

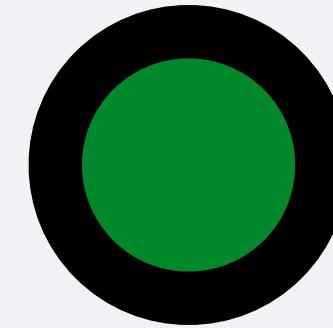
# Ветки



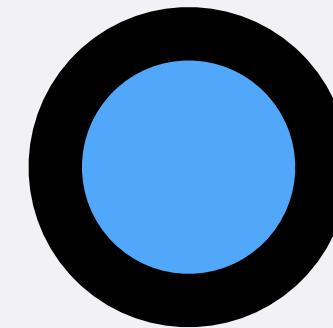
develop



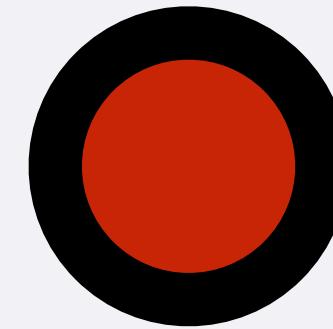
feature



release

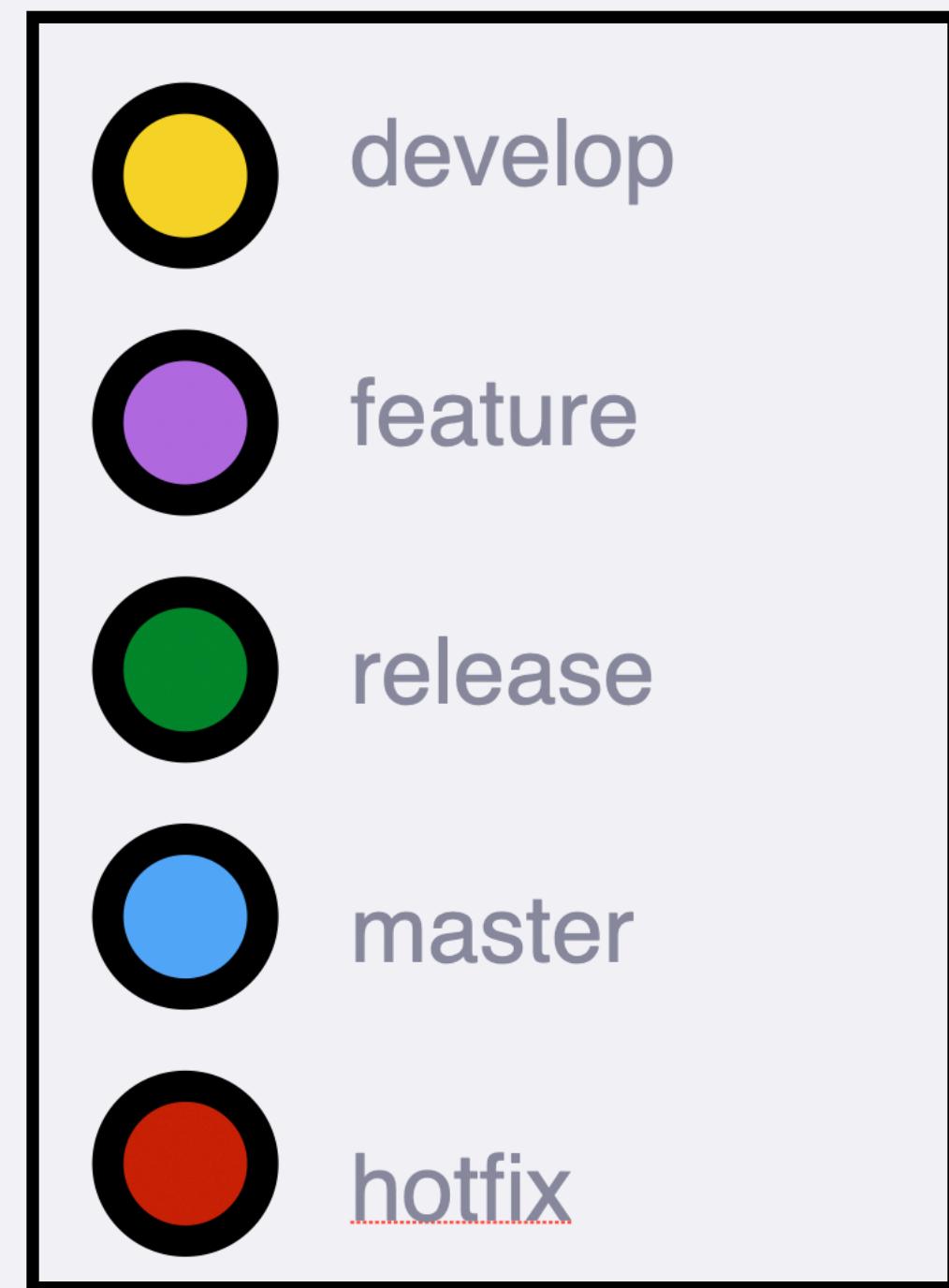
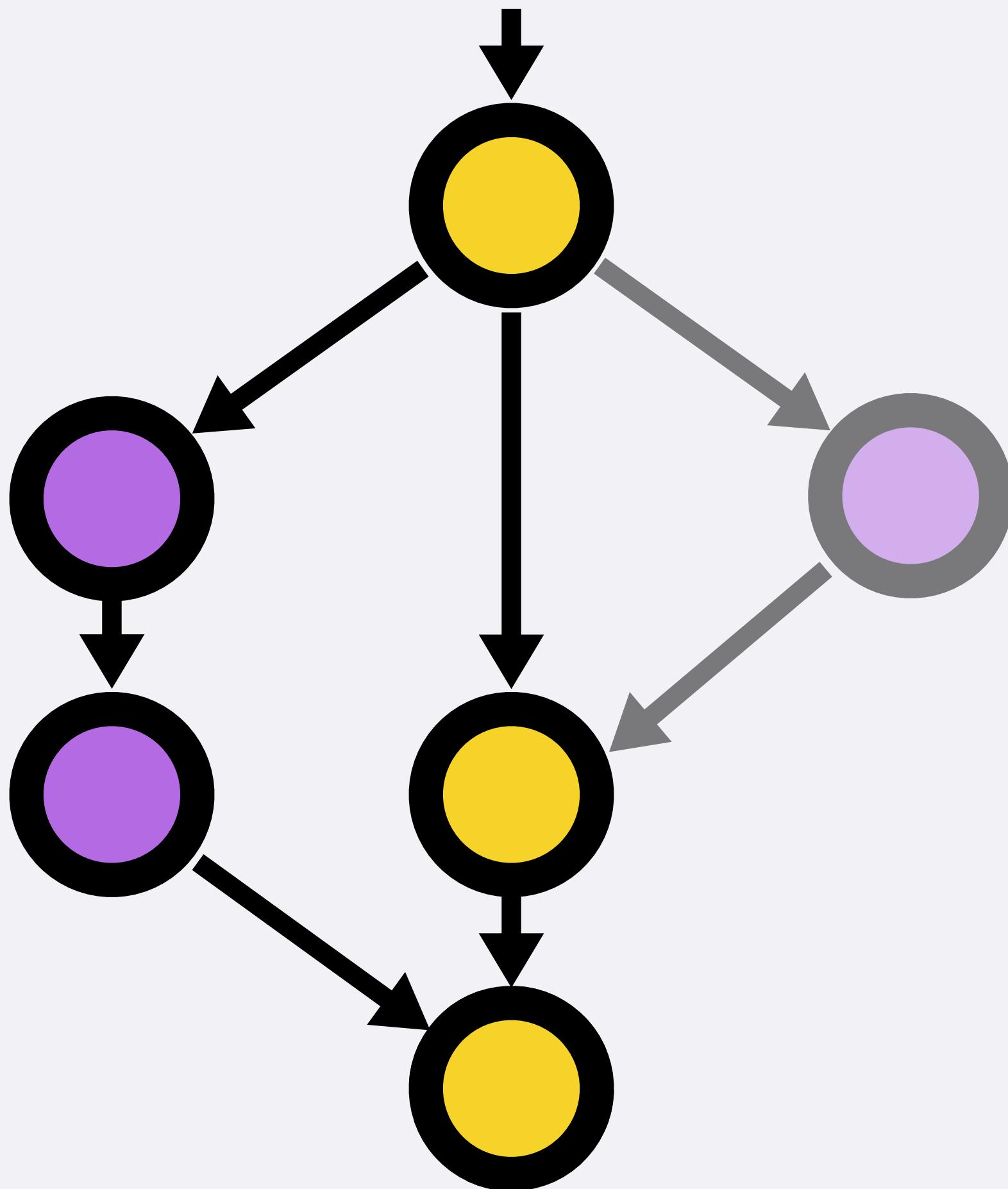


master

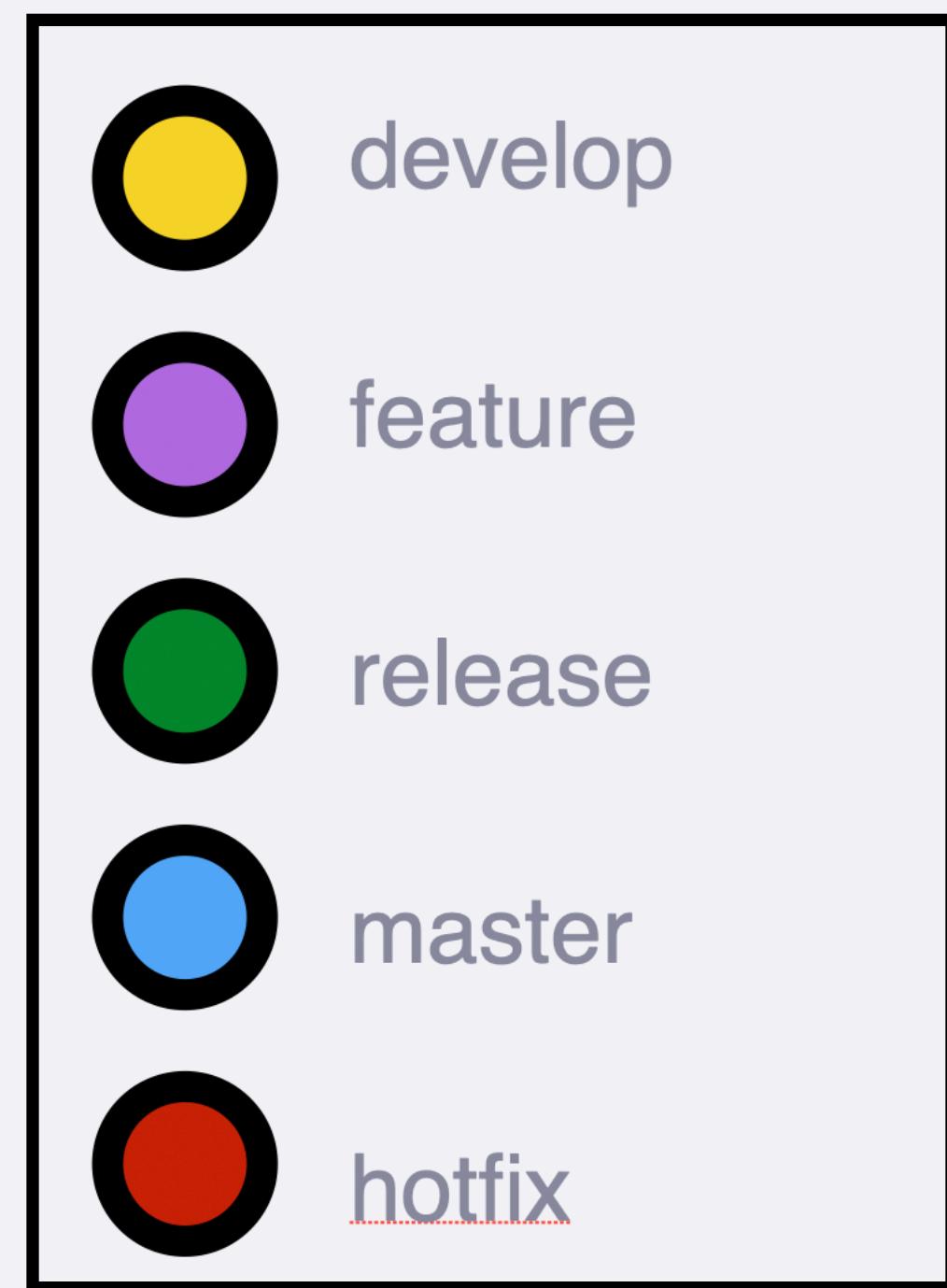
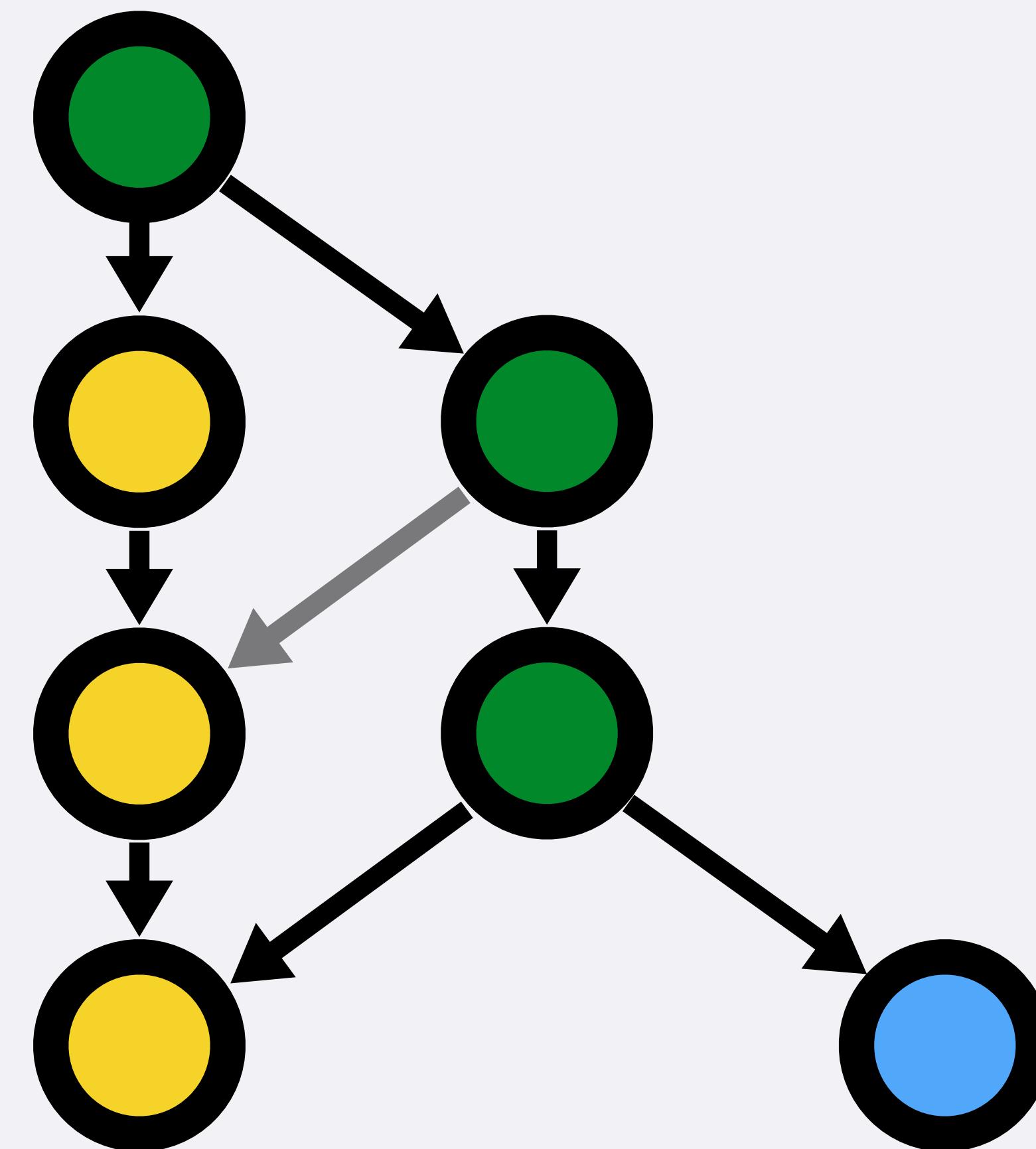


hotfix

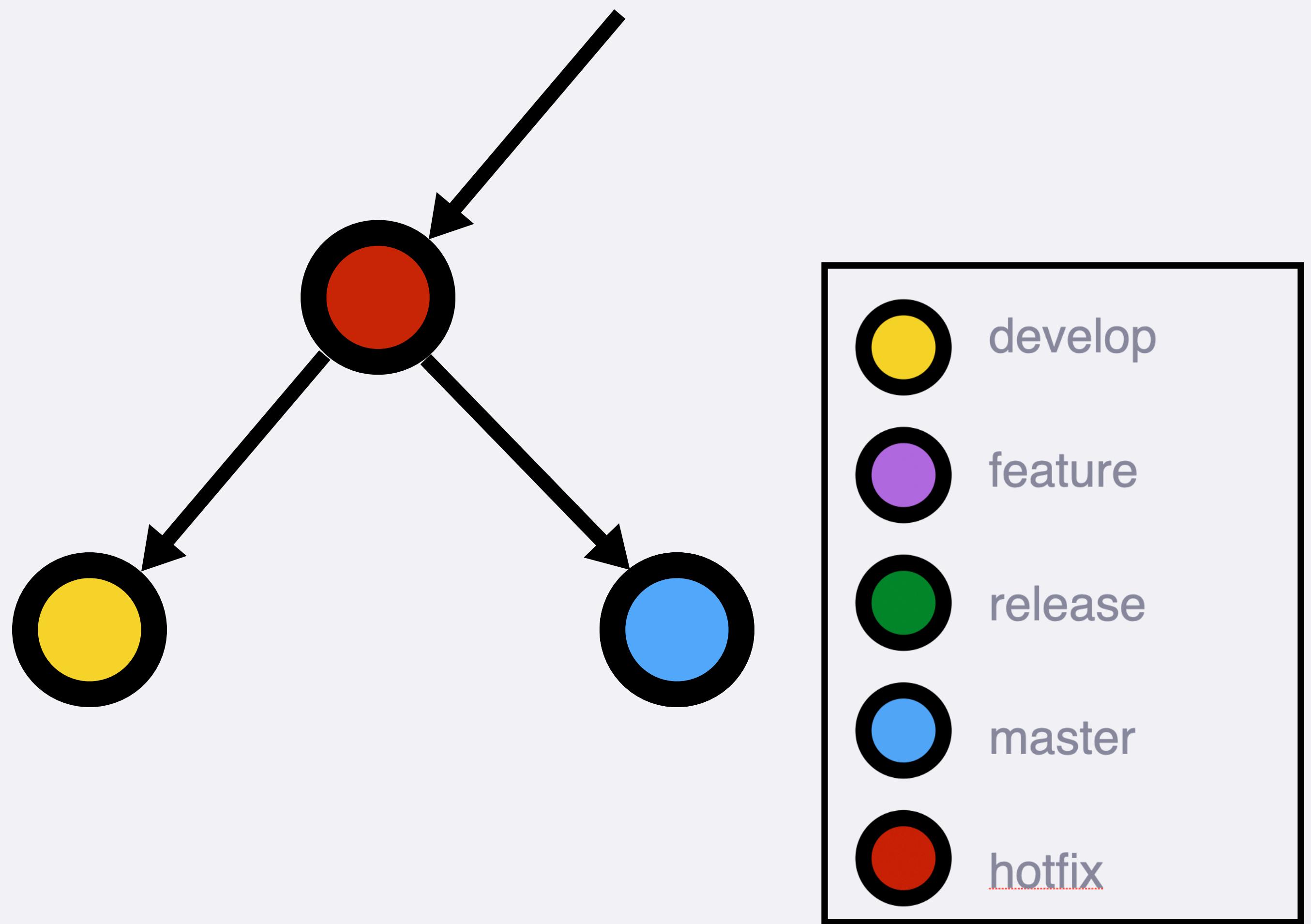
# Feature



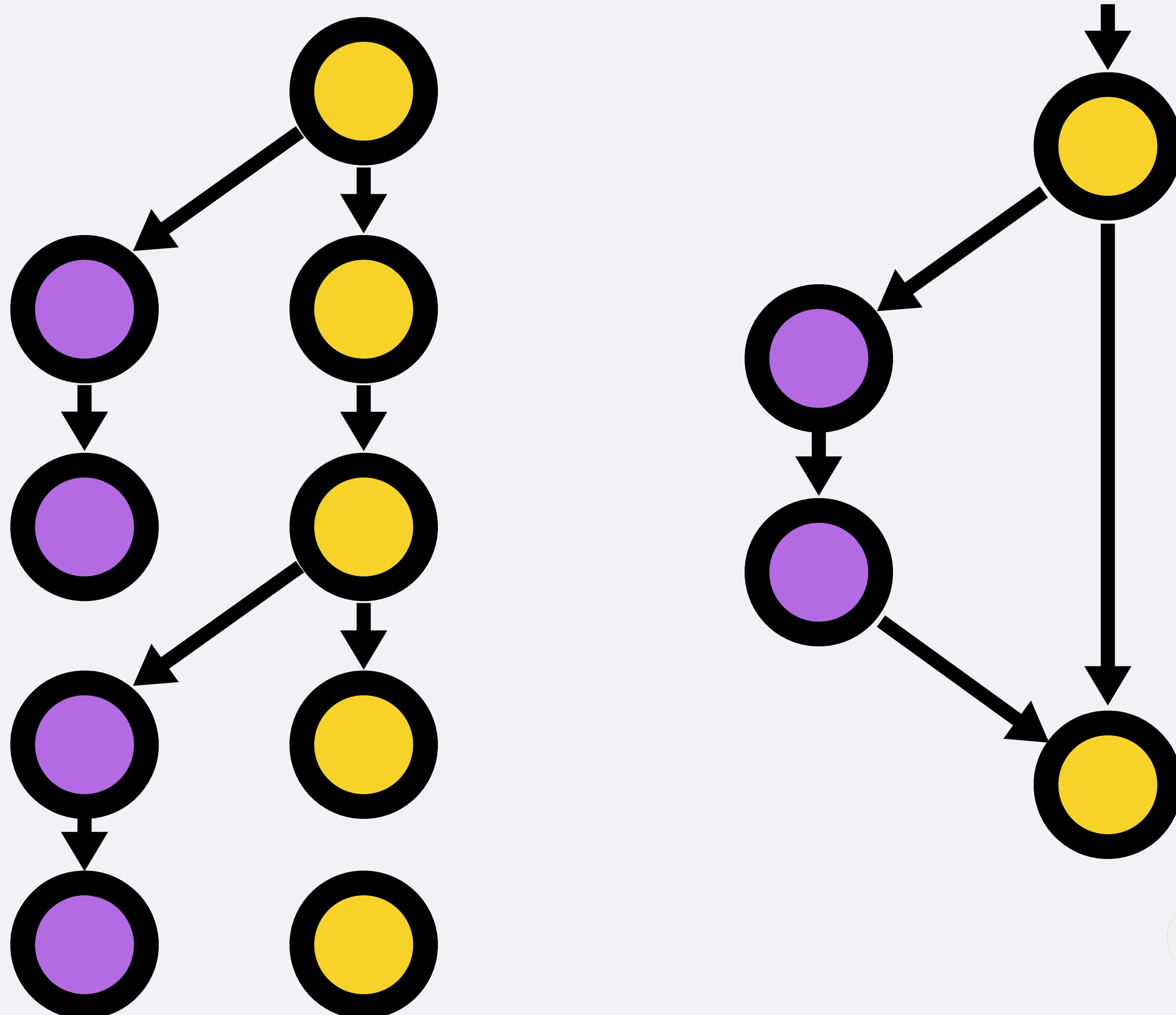
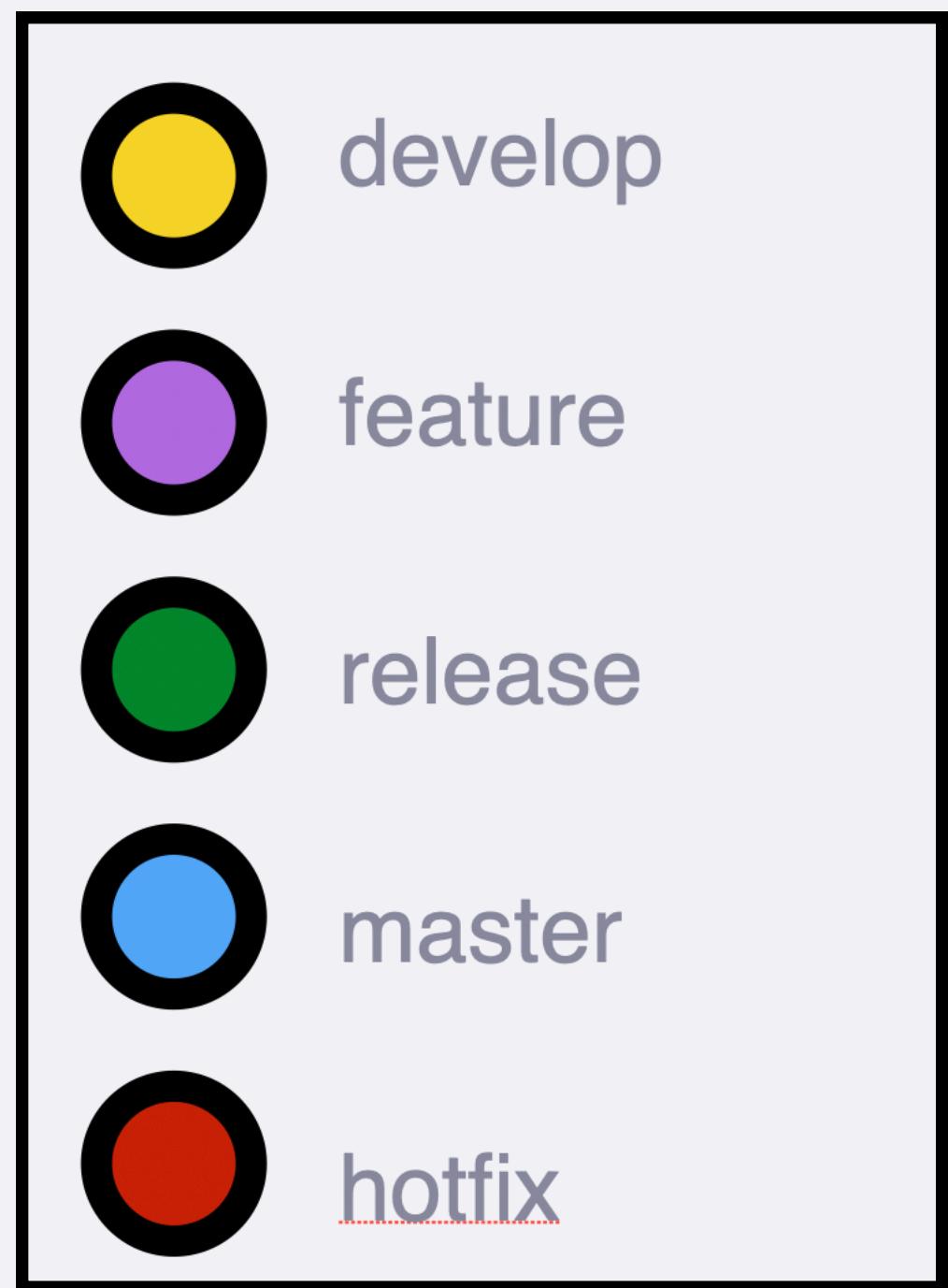
# Release



# Hotfix



# Rebase



Devteam

# Interactive Rebase

1

```
$ git checkout feature  
$ git rebase -i master
```

2

```
pick 22d6d7c Commit message#1  
pick 44e8a9b Commit message#2  
pick 79f1d2h Commit message#3
```

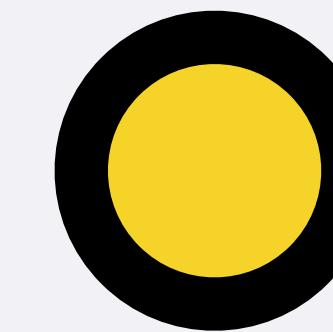
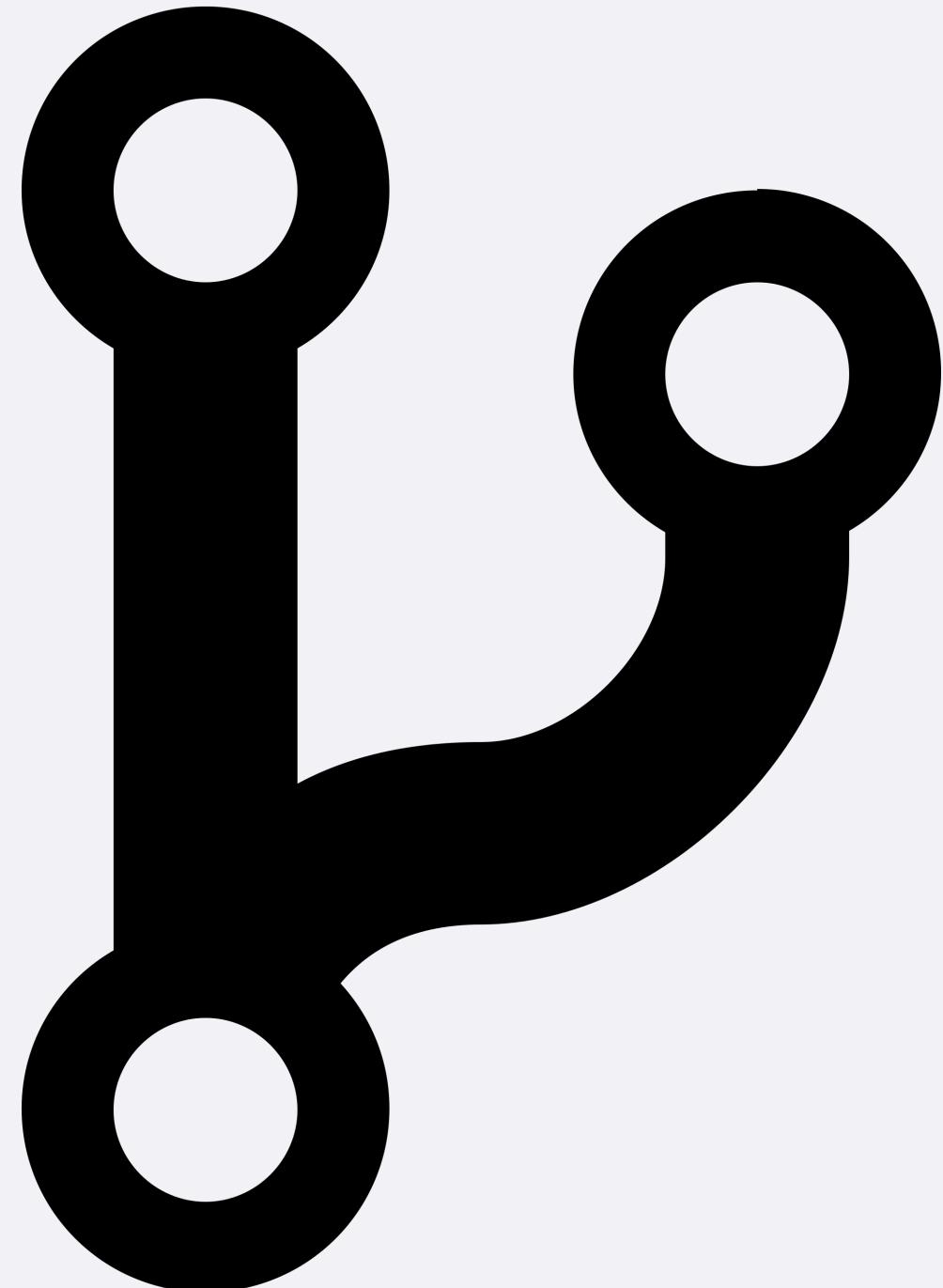
3

```
pick 6633b5a Message #1  
pick 3a03f0d Message #2  
pick 657897b Message #3  
  
# Rebase 3598fe2..657897b onto 3598fe2 (3 command(s))  
#  
# Commands:  
# p, pick = use commit  
# r, reword = use commit, but edit the commit message  
# e, edit = use commit, but stop for amending  
# s, squash = use commit, but meld into previous commit  
# f, fixup = like "squash", but discard this commit's log message  
# x, exec = run command (the rest of the line) using shell  
# d, drop = remove commit  
#  
# These lines can be re-ordered; they are executed from top to bottom.  
#  
# If you remove a line here THAT COMMIT WILL BE LOST.  
#  
# However, if you remove everything, the rebase will be aborted.  
#  
# Note that empty commits are commented out
```

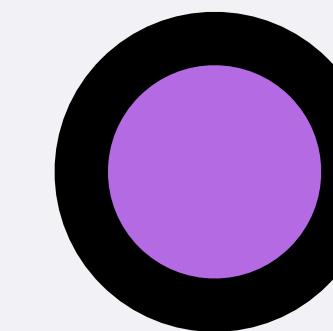
# Rebase vs Merge

- Перемещает ветку в нужное место
- Конфликты решаются покоммитно
- Красивая история
- Создается новый коммит
- Решаются все конфликты
- Настоящая история

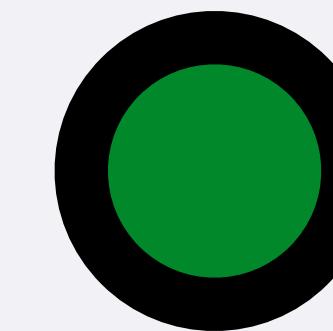
# Sberbank flow



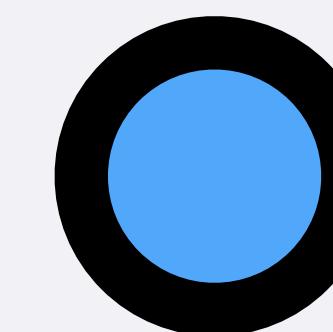
develop



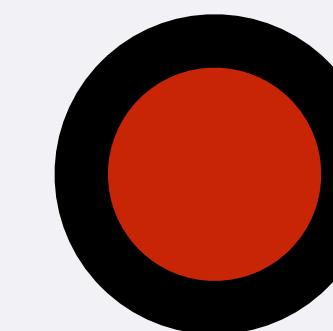
feature



release



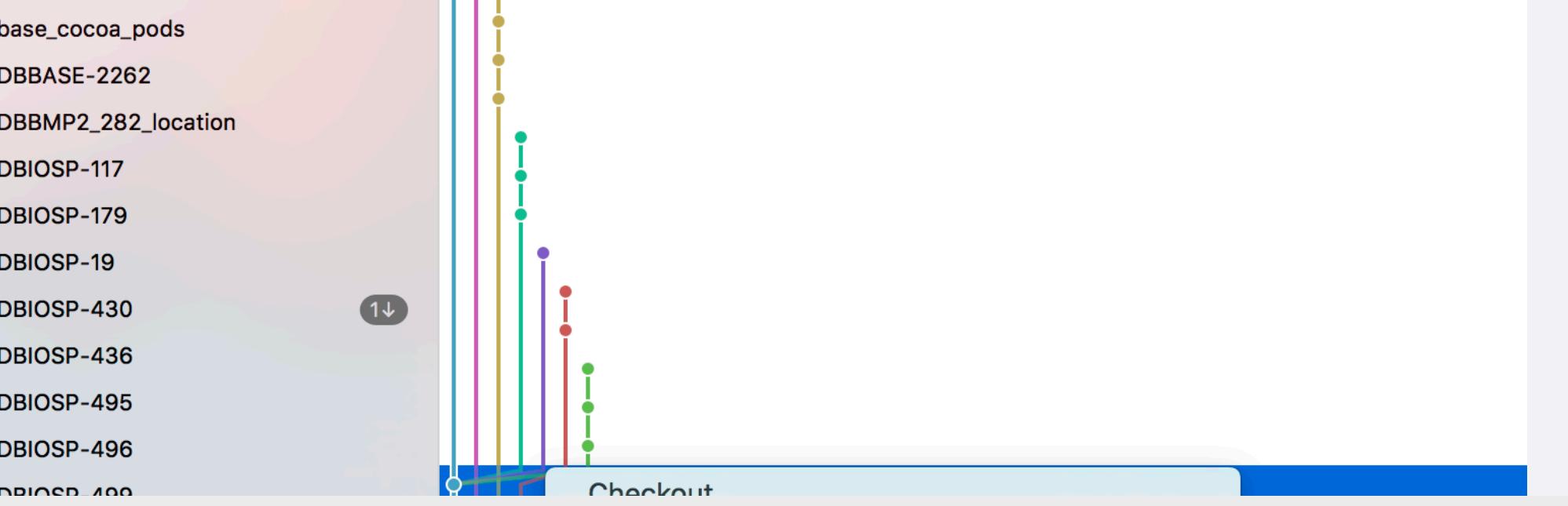
master



hotfix

# Sberbank flow

- Squash
- rebase изменений на ветку develop
- git push --force
- Создать Merge Request



base\_cocoa\_pods  
DBBASE-2262  
DBBMP2\_282\_location  
DBIOSP-117  
DBIOSP-179  
DBIOSP-19  
DBIOSP-430  
DBIOSP-436  
DBIOSP-495  
DBIOSP-496  
DBIOSP-499

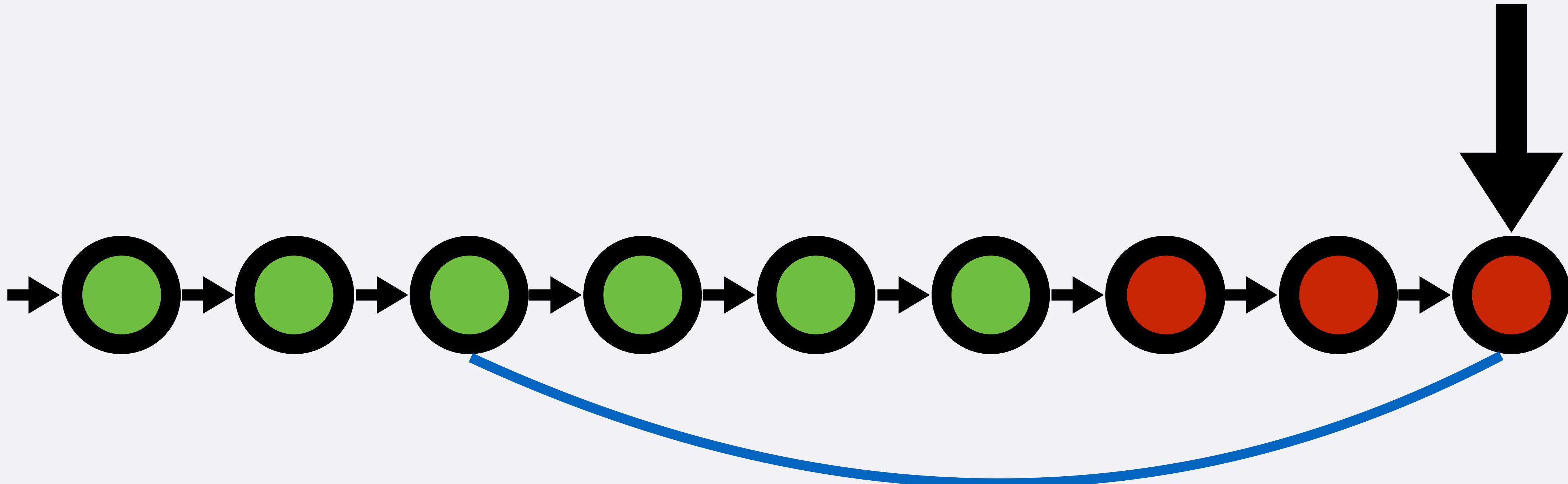
Reorder and amend commits: **ПРАВИЛЬНО**

Changeset	Amend Commit?	Description
▼ [3 commits]		DBIOSP-586 Разделение тестов на NSNumber и NSDecimalNumber (+2 squashed commits)
281b7d0		DBIOSP-586 Разделение тестов на NSNumber и NSDecimalNumber
aed460b		DBIOSP-586 Тесты на NSNumber
a037cd1		DBIOSP-586 Тесты на категорию NSCharacterSet
4f76e5b	<input type="checkbox"/>	DBIOSP-587 Удаление неиспользуемого метода
007d248	<input type="checkbox"/>	DBIOSP-587 Замена метода конвертации в base64 строку на стандартный
c769518	<input type="checkbox"/>	DBIOSP-587 Замена метода конвертации NSData из base64 строки на стандартный.
b87575e	<input type="checkbox"/>	DBIOSP-unitests Тесты на SBFoundation
1751c9d	<input type="checkbox"/>	DBIOSP-unitests Вынос категории NSArray + тесты

# Tips & Tricks

- Bisect
- Stash
- Patch
- Cherry Pick
- Amend
- git and storyboards

# Bisect



# Stash



- Локальное хранение изменений
- Вне структуры репозитория

# Patch



- Не связан с ветками
- Набор изменений

# Cherry pick



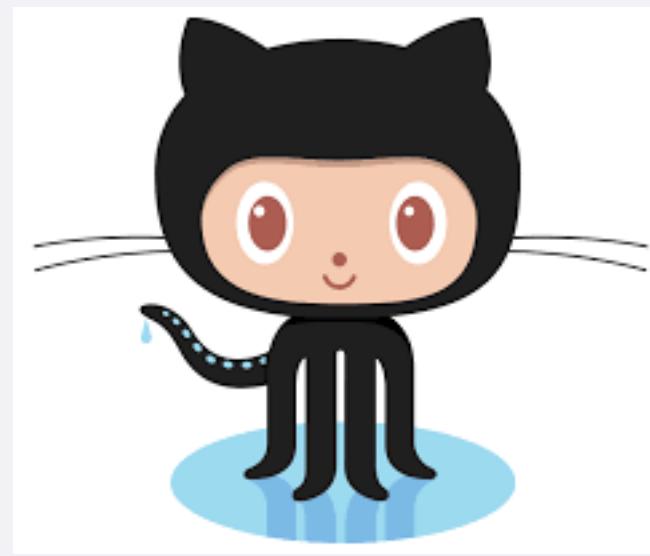
Копирование коммитов в  
свою ветку

# Amend



- Меняет последний коммит
- Только для локальных

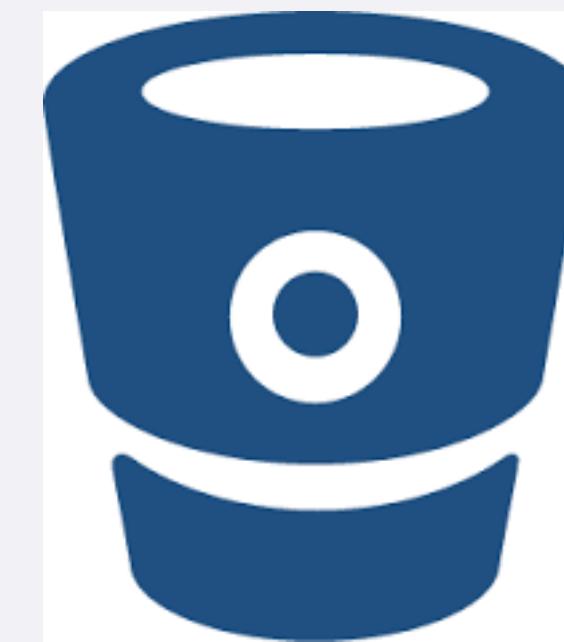
# Менеджеры репозиториев



GitHub



GitLab



Bitbucket

# Авторизация

- Username/password
- ssh-key

# Git and storyboards

- SwiftUI
- Отказ от storyboards
- Diff instruments



# Почитать

1. <https://git-scm.com/book/en/v2>
2. <https://learngitbranching.js.org/>
3. <https://github.com/Lemonbrush/Contribute-To-This-Project>
4. <https://nvie.com/posts/a-successful-git-branching-model/>
5. <https://choosealicense.com/>
6. <https://github.com/bk2204/git/blob/transition-stage-4/Documentation/technical/hash-function-transition.txt>
7. <https://medium.com/flawless-app-stories/ios-storyboard-merge-hell-b4cbb2e57dfc>

# Домашнее задание

- Зарегистрироваться на GitHub
- Поставить Fork <https://fork.dev/>
- Создать проект (допустимо несколько) под домашние задания
  - добавить readme
  - добавить лицензию
- **Эссе:** описать отличия Trunk-based development от feature-подхода

**Срок: до 25-го июня**

# Trunk-based development

1. <https://trunkbaseddevelopment.com/>
2. <https://habr.com/ru/post/519314/>

# Additional reading

1. <https://martinfowler.com/bliki/CodeOwnership.html>
2. <https://semver.org/>