

GIT – теория и сценарии использования

Групповая динамика и коммуникации в профессиональной практике программной инженерии

Кузнецов Михаил Александрович
ma.kuznetsov@hse.ru
@rokymiel



ФКН

факультет компьютерных наук

2025

Кузнецов Михаил



Старший iOS разработчик
Т-Банк Поиск (~ 5 лет)



Магистр программной
инженерии



Выпускник ВШЭ ПИ 2023



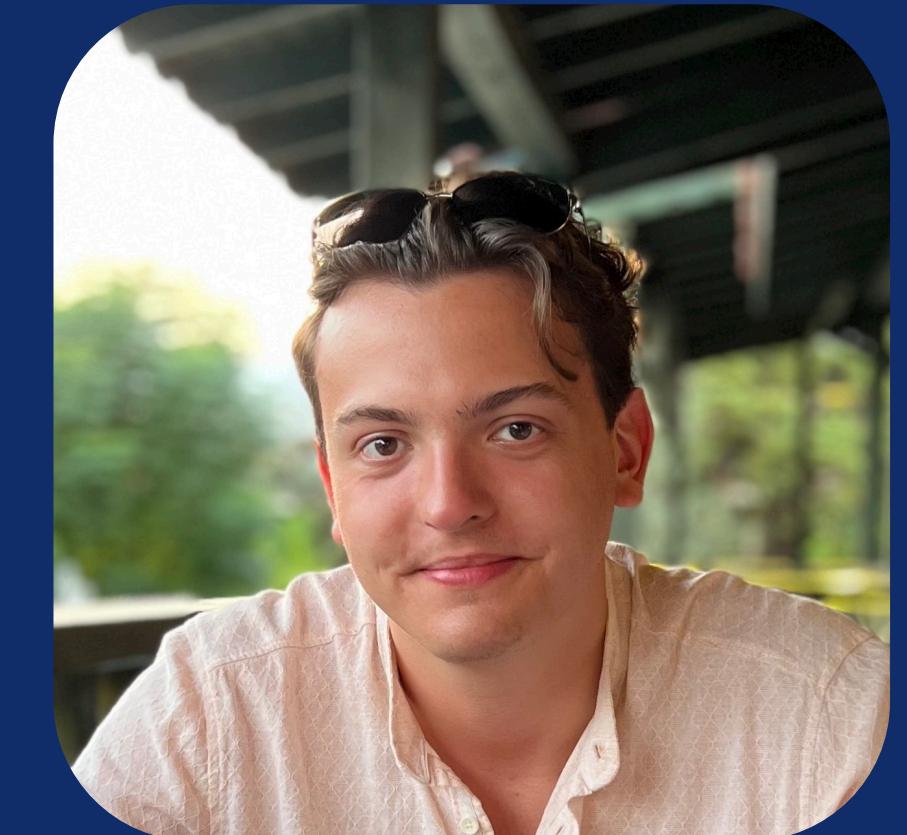
ROKYMIEL



Преподаватель
ВШЭ & ЦУ & Т-Образования



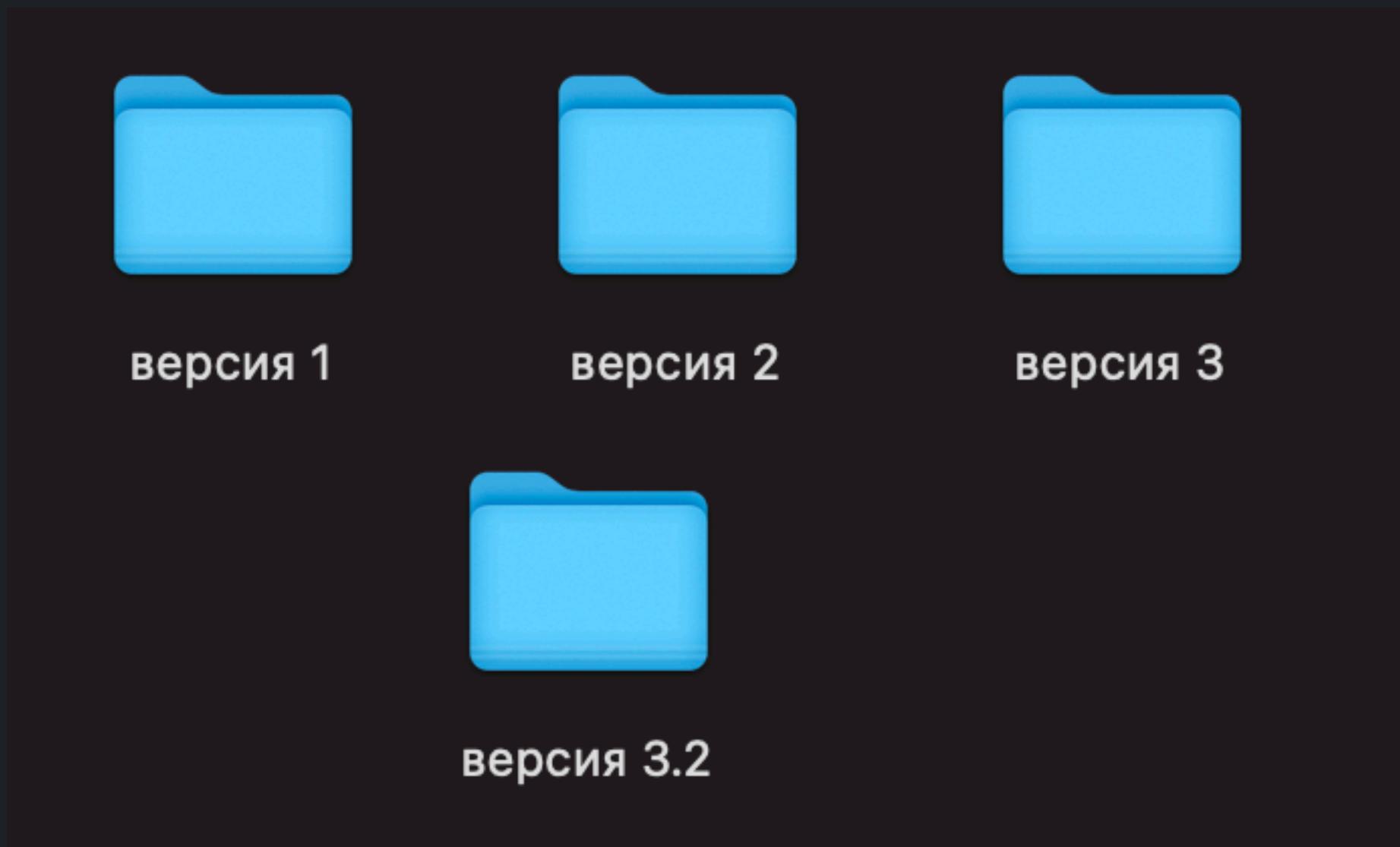
Руководитель трека
«Мобильная разработка»
PROD



GIT – команды

Групповая динамика и коммуникации в профессиональной практике
программной инженерии

Проекты раньше



Ситуация

Ситуация

Петя просит Васю:

Ситуация

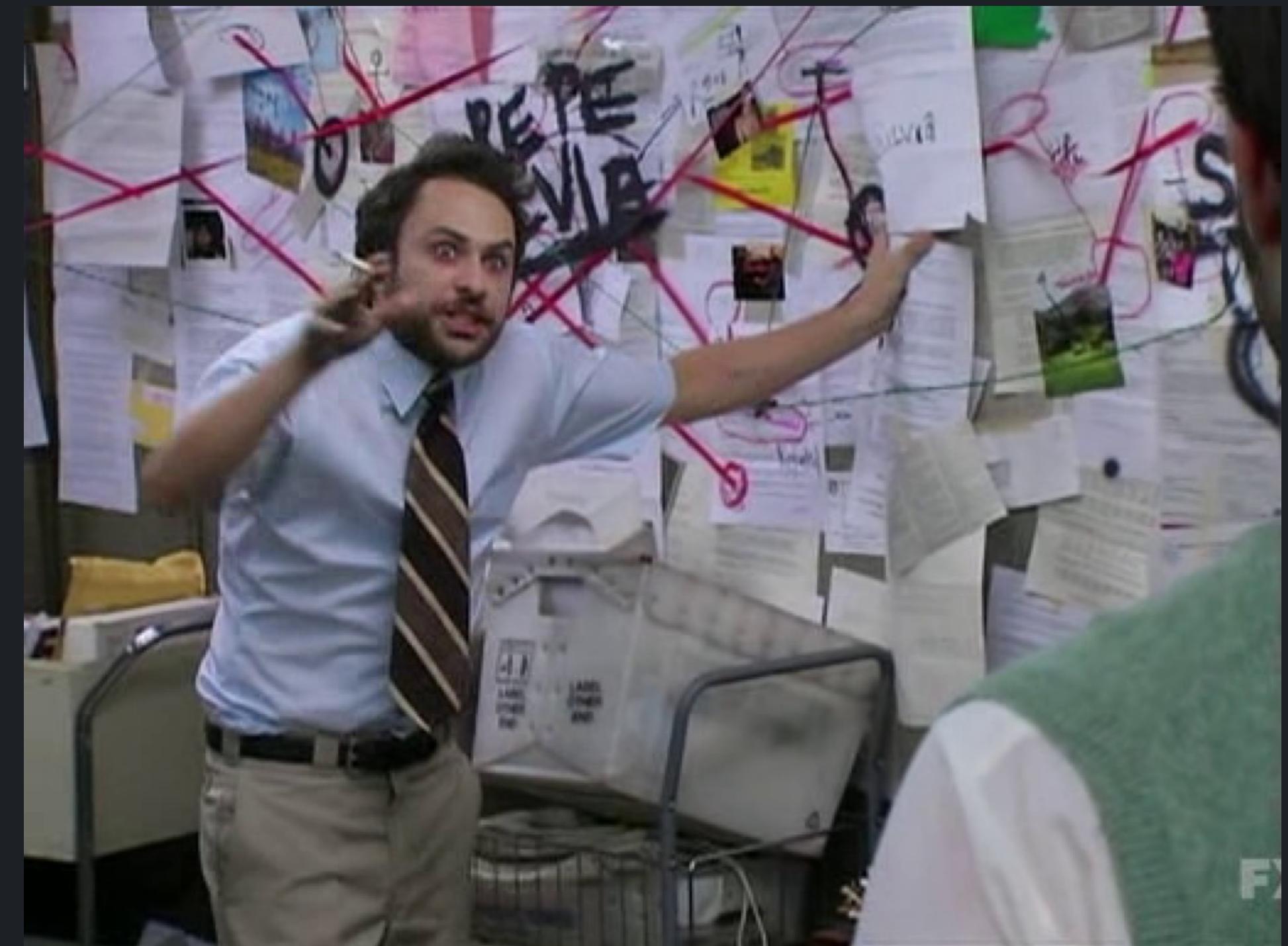
Петя просит Васю:

- добавить в **версию 3.0** текстовое поле из **версии 5.5**, чтобы получить **версию 3.1**

Ситуация

Петя просит Васю:

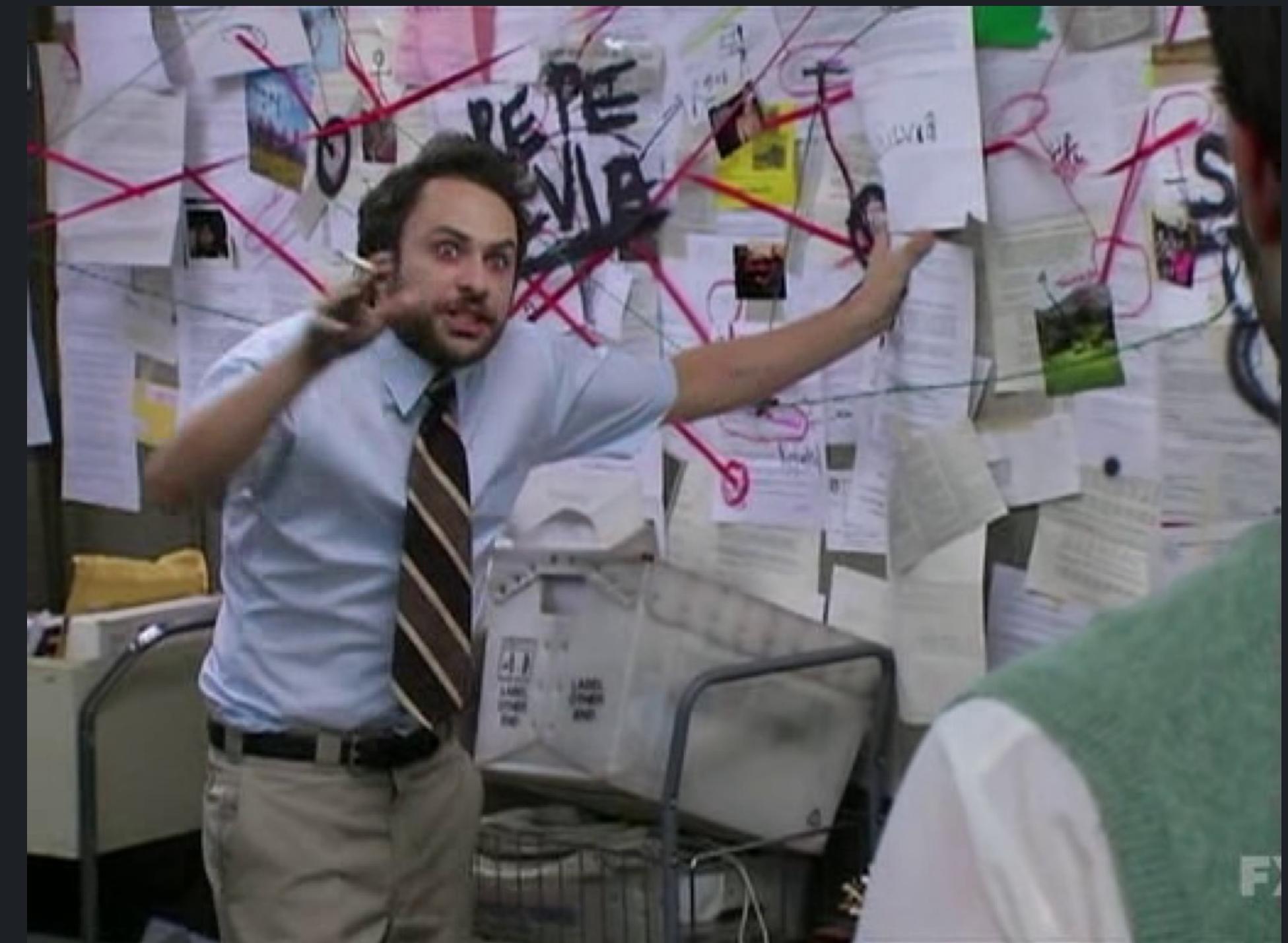
- добавить в **версию 3.0** текстовое поле из **версии 5.5**, чтобы получить **версию 3.1**
- удалить из **версии 5.0** кнопку "Оплатить позже"



Ситуация

Петя просит Васю:

- добавить в **версию 3.0** текстовое поле из **версии 5.5**, чтобы получить **версию 3.1**
- удалить из **версии 5.0** кнопку "Оплатить позже"
- найти в **версии 5.5** как было сделана отправка на e-mail



Для чего нужен Git?

- Управления версиями кода
 - Отслеживание состояния файлов
 - Именование версий
- Совместной работы над кодом
 - Одновременная работа над кодом
 - Независимость разработчиков

Git

Git

✓ Система контроля версий (VCS)

Git

- ✓ Система контроля версий (VCS)
- ✓ Распределенная

Git

- ✓ Система контроля версий (VCS)
- ✓ Распределенная
- ✓ Гибкая, производительная

Терминология Git

- Состояние (commit)
- Подсчет разницы (diff)
 - ! хранится не все состояние, а разница с предыдущим для экономии
- Легко запоминаемые состояния (branch, tag)

Commit

разница с предыдущим состоянием

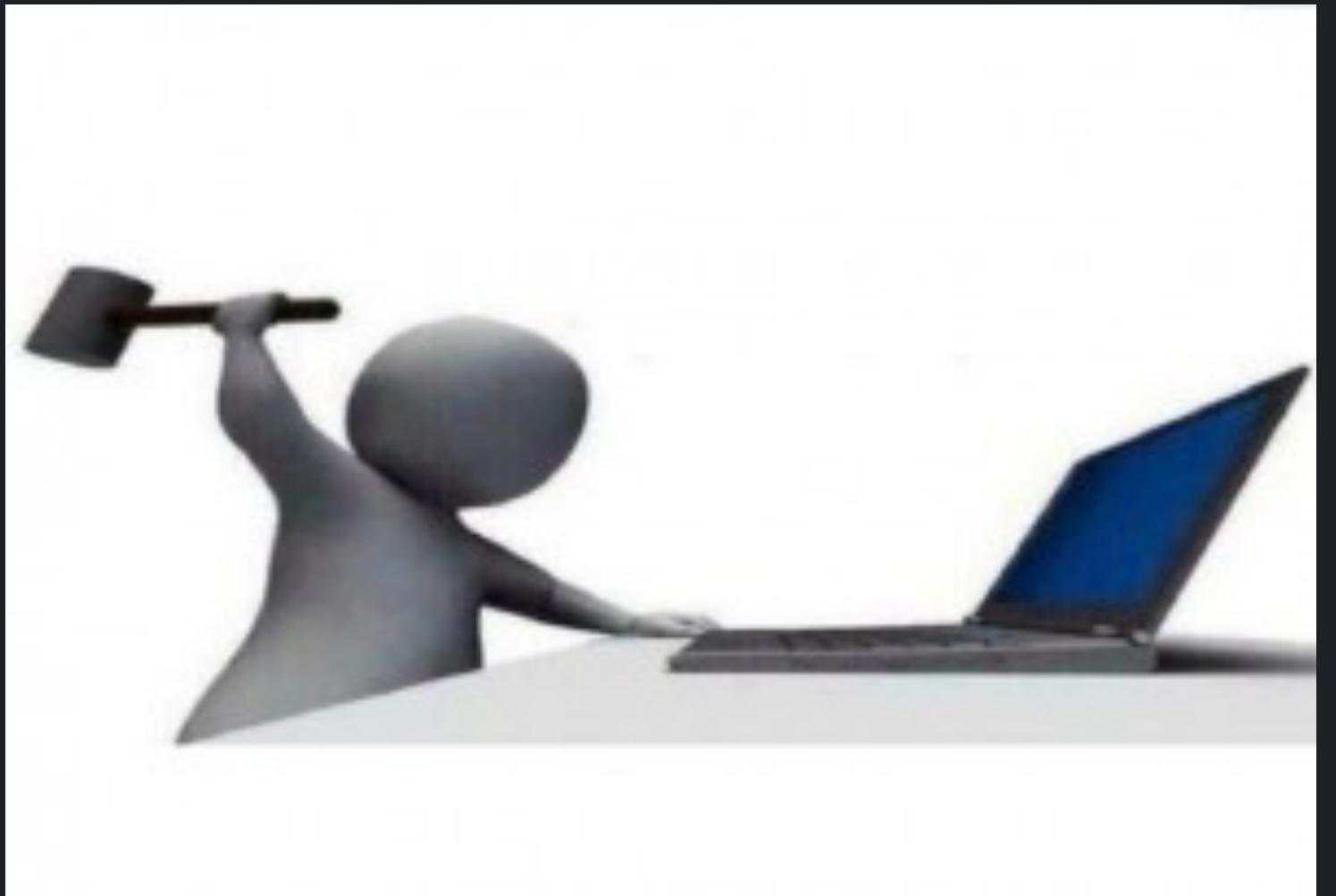
+ автор, временная метка

+ название

+ ссылка на предыдущий

= SHA-сумма

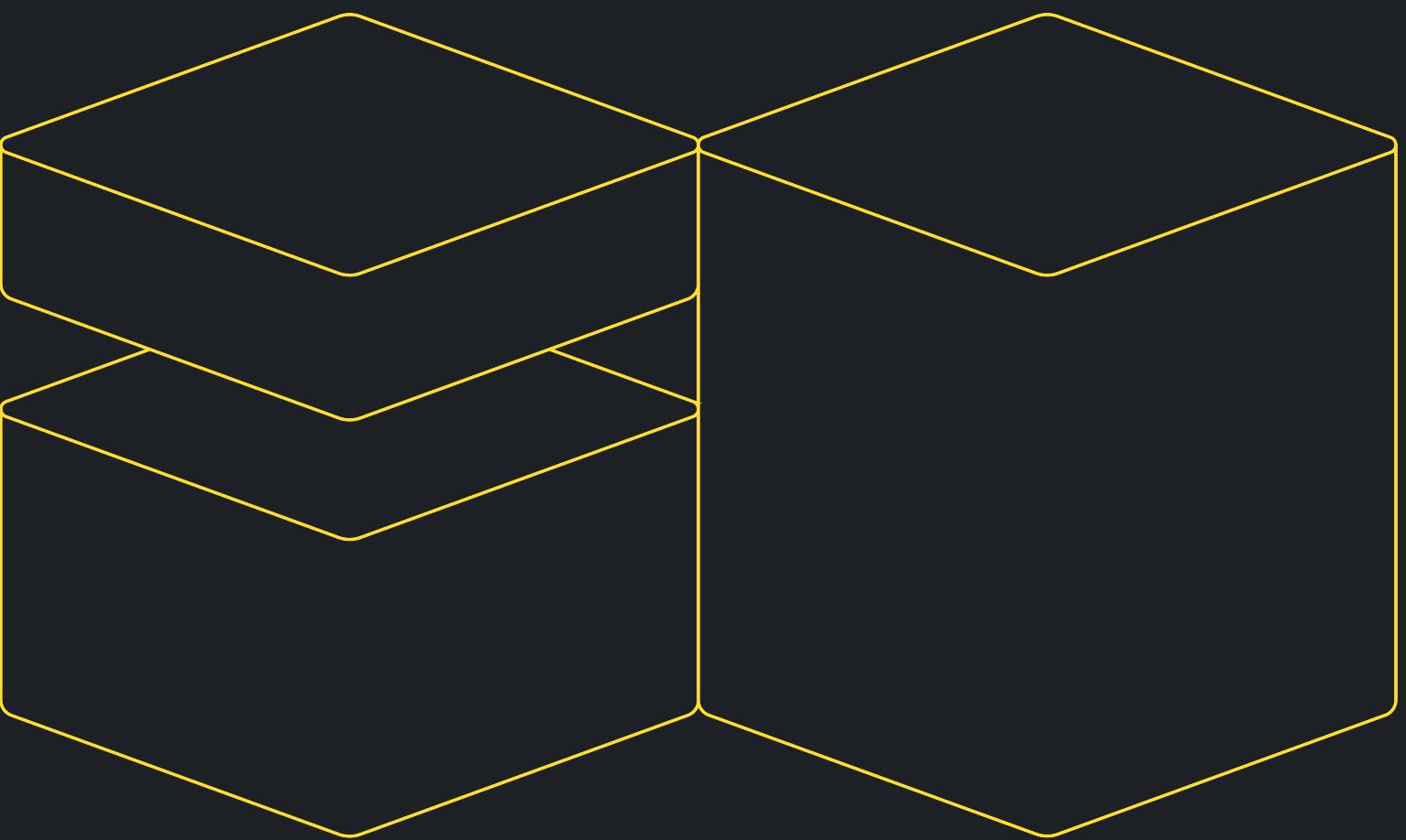
✓ уникальный идентификатор состояния



Подсчет разницы

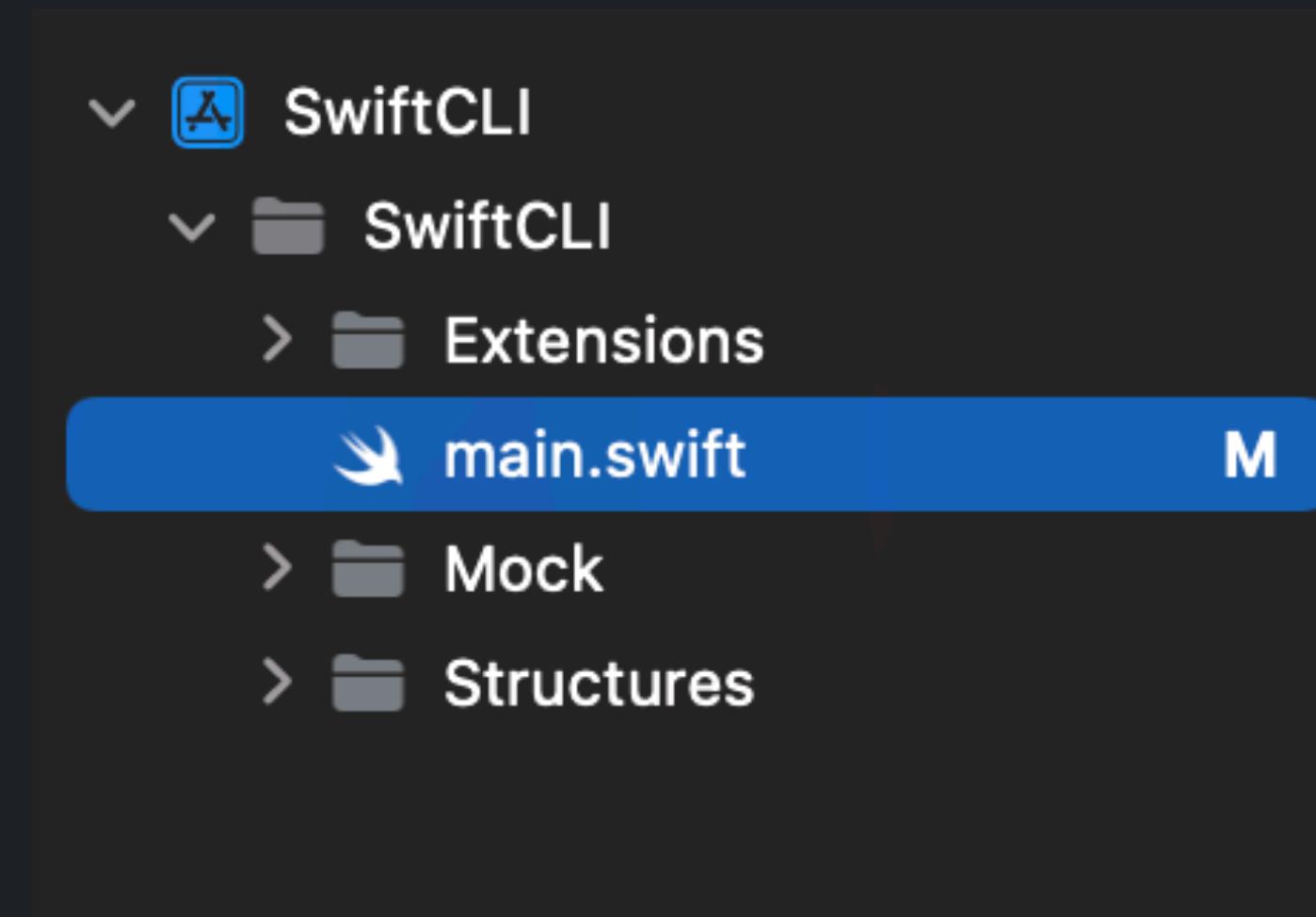
Типы файлов

- Текстовые файлы
- Бинарные файлы



Текстовые файлы

- .swift
- .xib
- .json



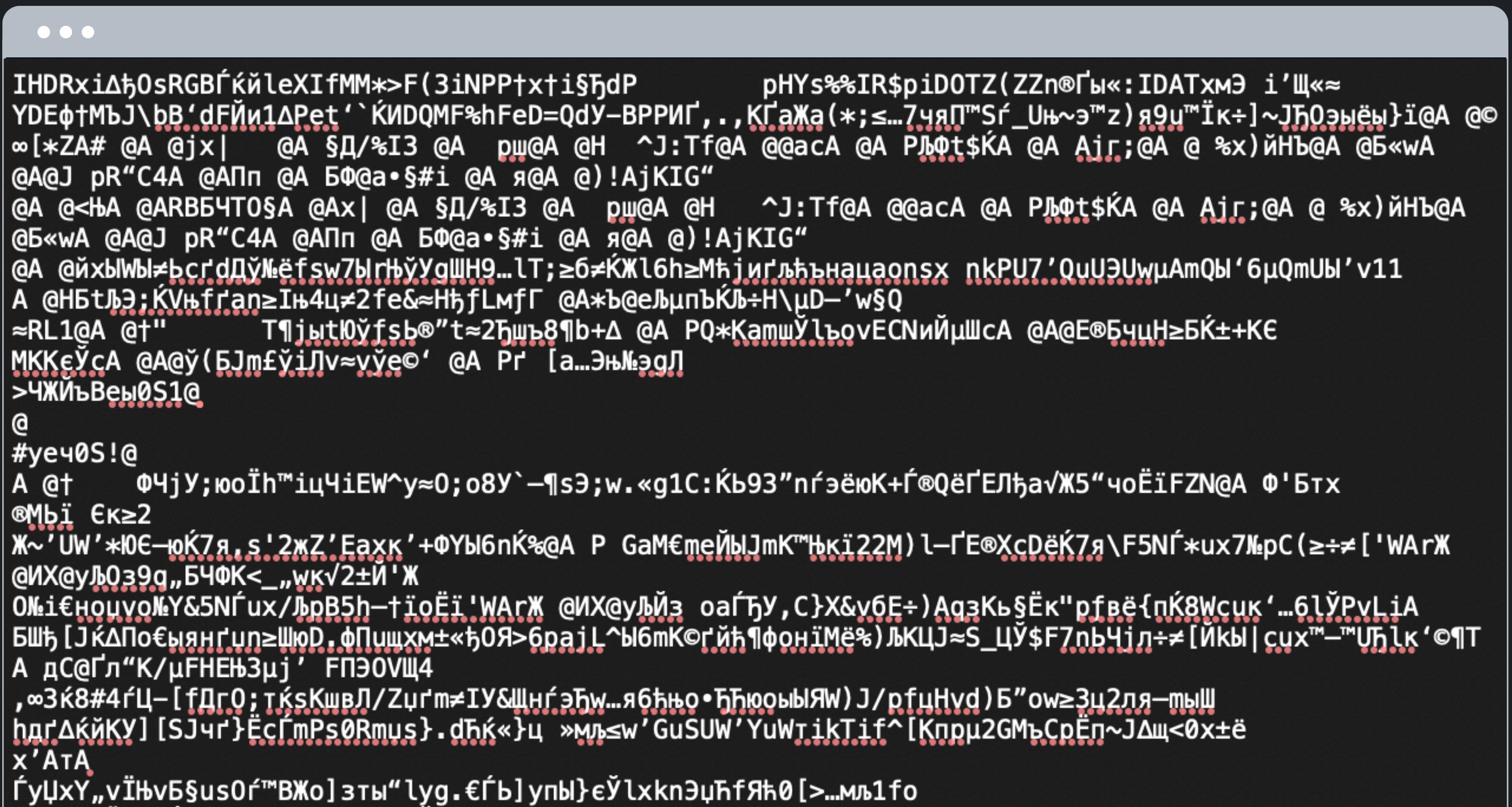
The screenshot shows a file browser interface with a dark theme. On the left, there's a tree view of a project structure:

- SwiftCLI (selected)
- SwiftCLI (unselected)
- Extensions (unselected)
- main.swift (selected, highlighted with a blue bar)
- Mock (unselected)
- Structures (unselected)

On the right, the content of the selected file, main.swift, is displayed in a code editor:

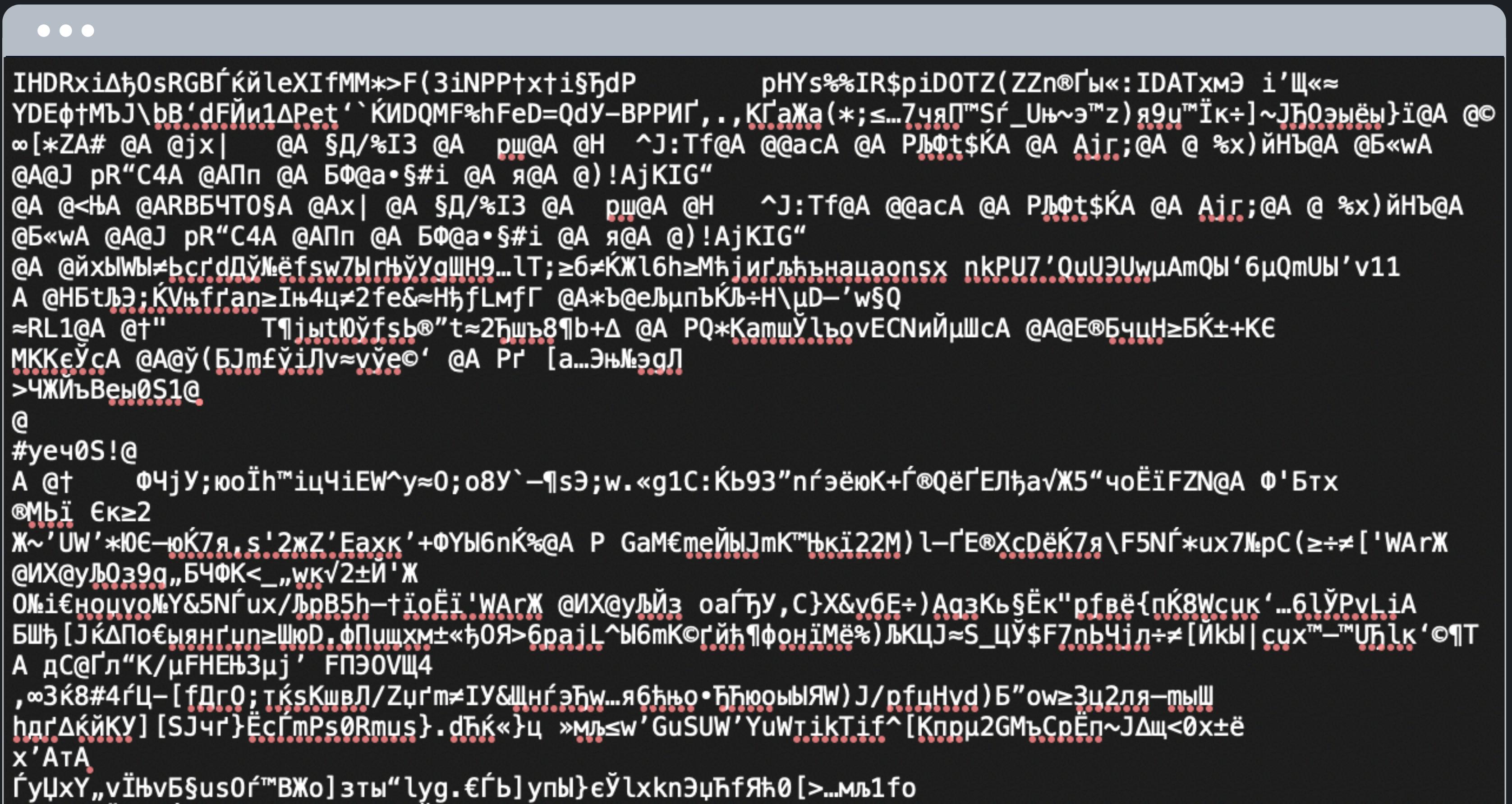
```
import Foundation
print("Hello, Tinkoff!")
```

Бинарные файлы



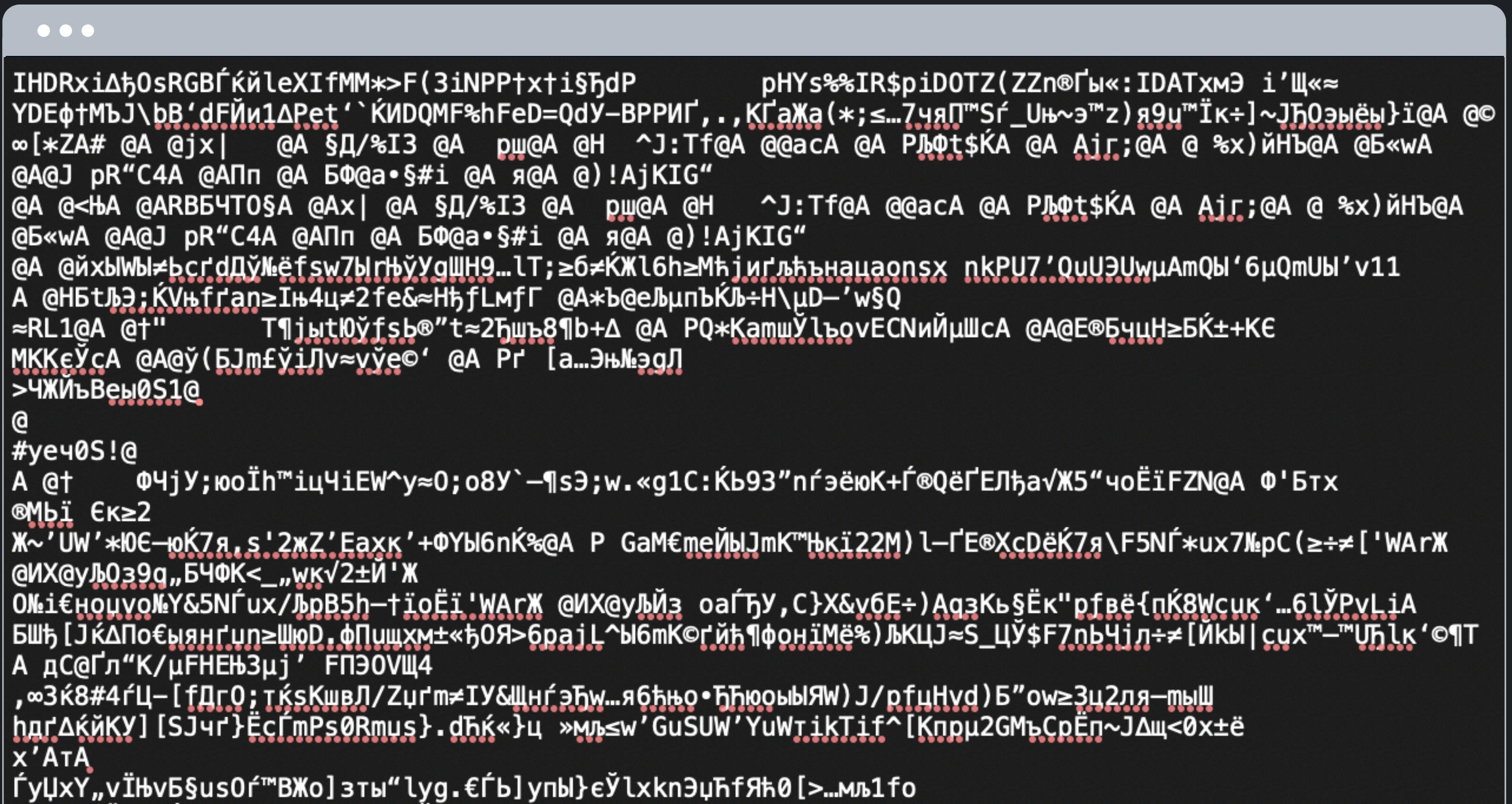
Бинарные файлы

- .jpg, .png, .mov



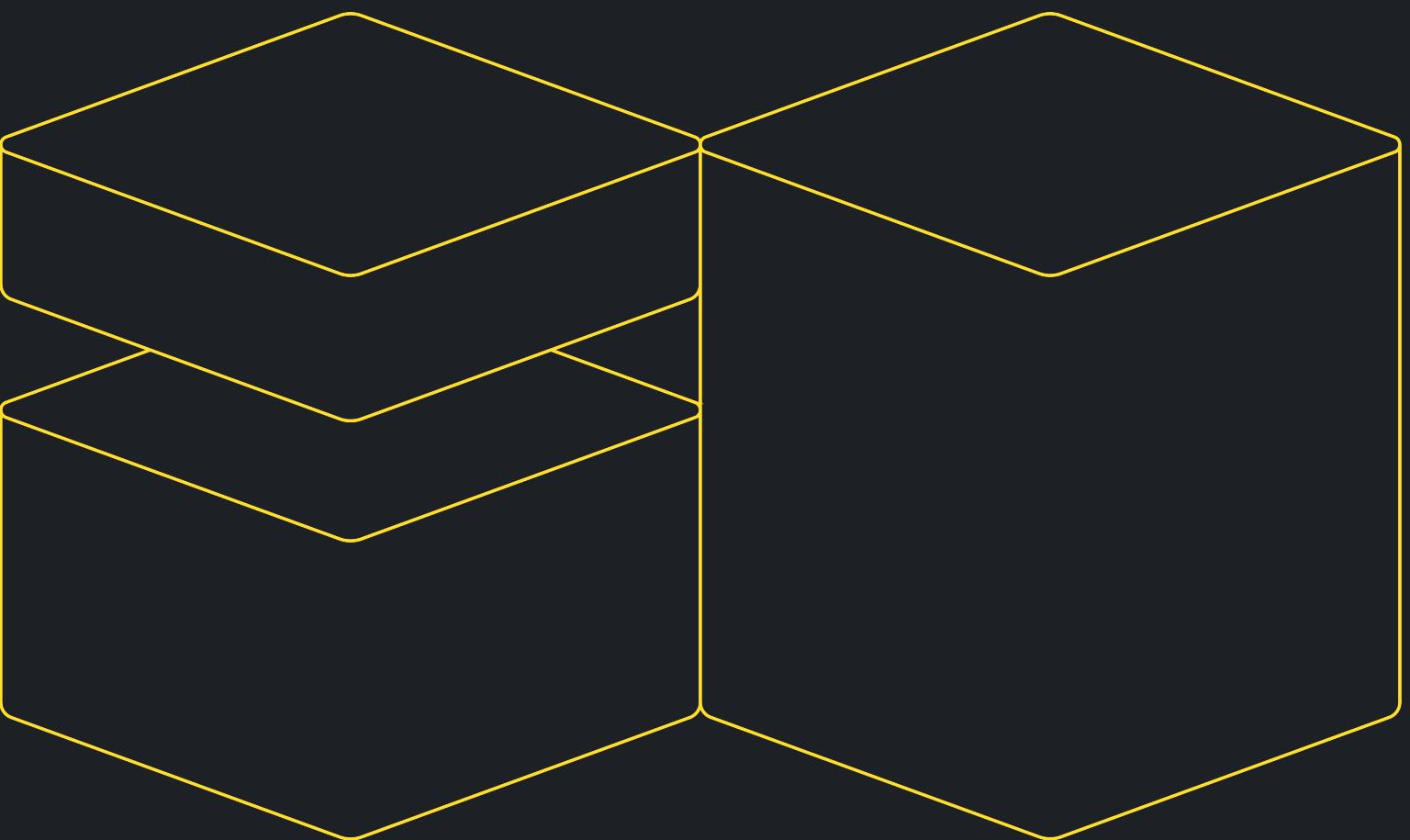
Бинарные файлы

- .jpg, .png, .mov
- .pdf



Механизм Diff

- Текстовые:
 - добавленные строки
 - удаленные строки
- Бинарные:
 - удаленные, добавленные
 - ⚠ невозможно подсчитать разницу, хранятся целиком



Механизм Diff

Механизм Diff

```
import Foundation
```



```
import UIKit  
print("Hello, T-Bank! »)
```

Механизм Diff

import Foundation



import UIKit
print("Hello, T-Bank!»)

```
@@ -1 +1,2 @@
-import Foundation
+import UIKit
+print("Hello, T-Bank!»)
```

Базовые команды

С чего начать?

- `git help COMMAND`
- `git status`
- Ситуация: "потерялся" и не знаешь что делать

Создание репозитория

- `git init`
- Директория `.git`
- Ветка по умолчанию `dev` (`main`, `master`)
- Ситуация: начинаем новый проект

Добавление файлов

- Указать файл(ы) или путь к ним
- Пример, `git add FILE_PATH`
- Ситуация: создаю новый экран в приложении

Удаление файлов

- `git rm FILE1 FILE2`
- `--cached` сохранит файл на диске
- Ситуация: удаляю старый экран с приложения

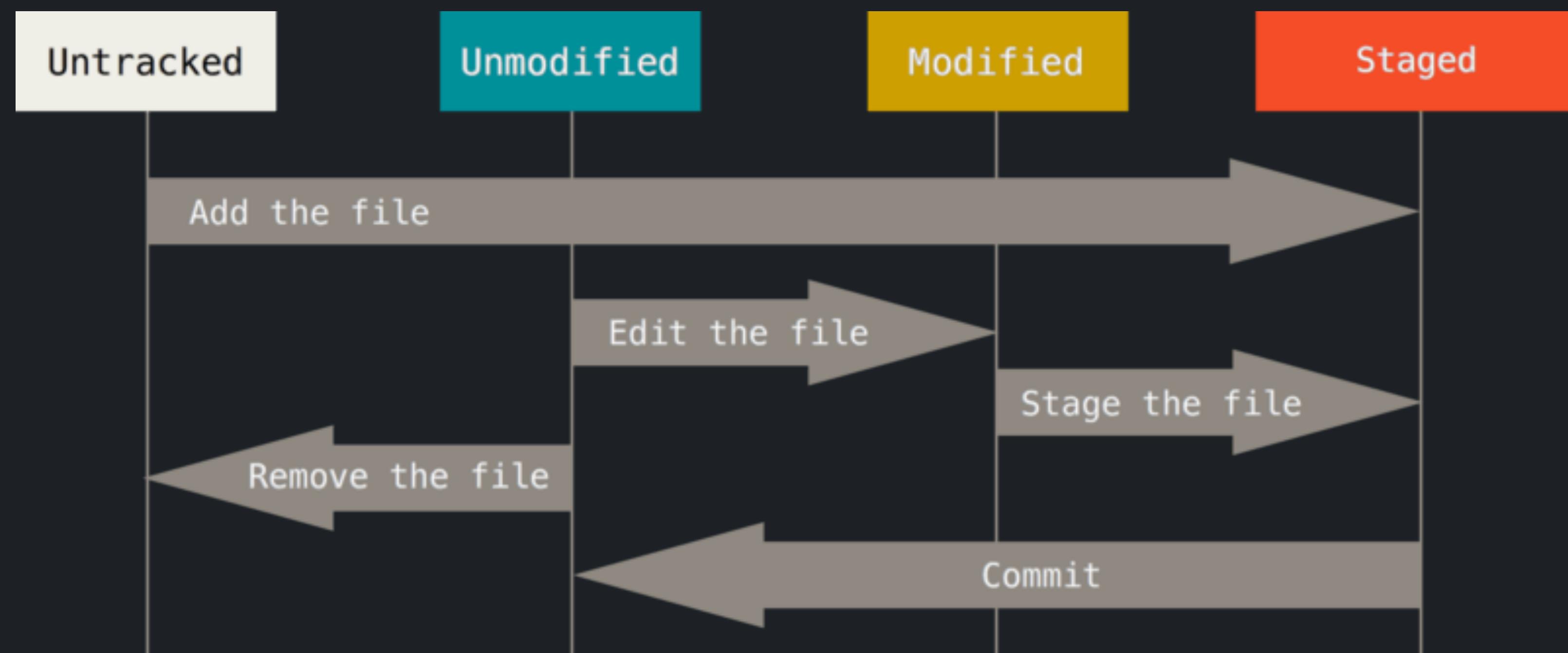
Игнорирование файлов

- .gitignore
- Лежит в корне проекта
- 🤯 можно добавить в git
- Ситуация: игнорирую build в папке проекта



Состояния файлов

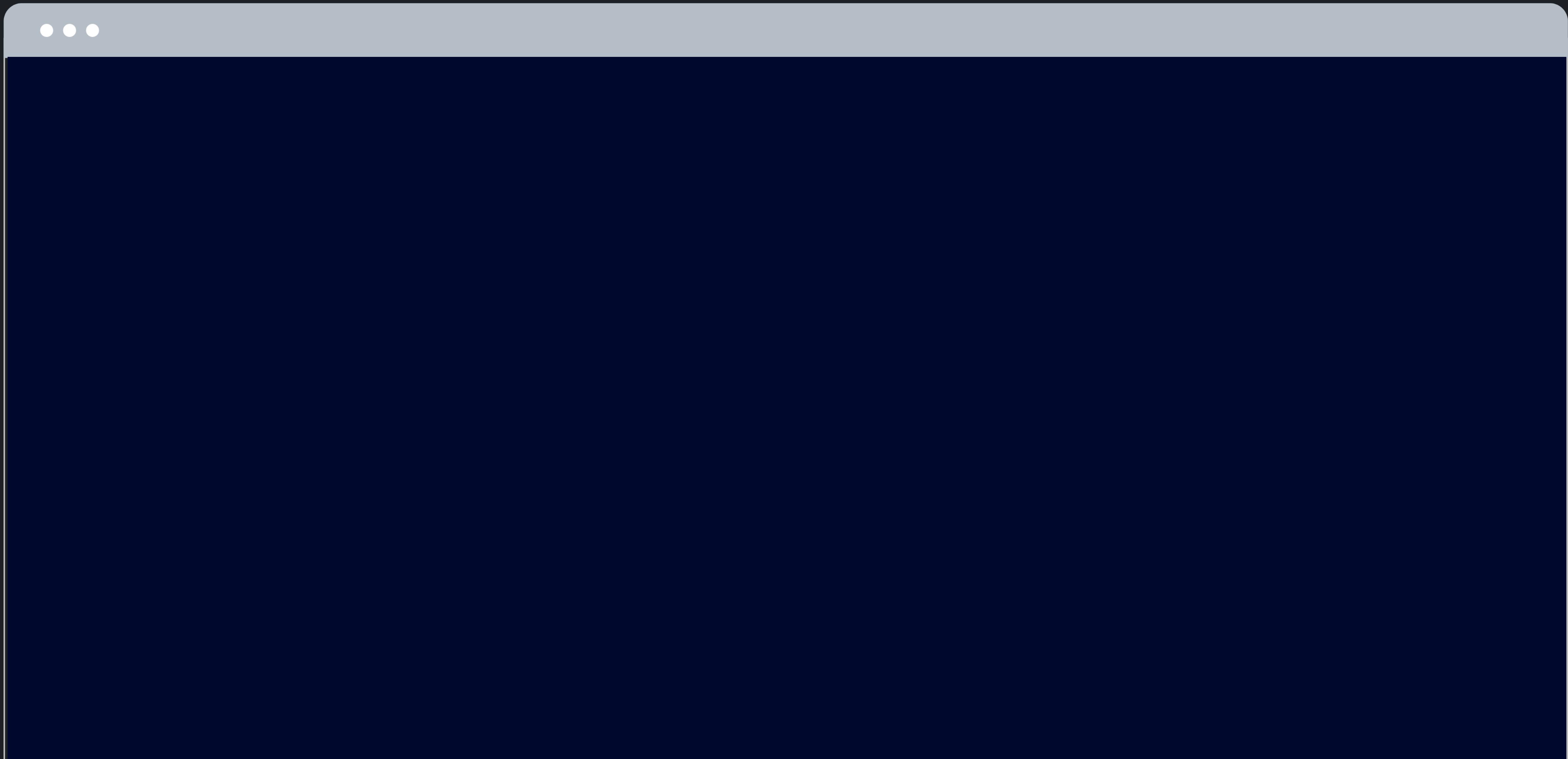
Состояния файлов



Commit

- `git commit -m "NAME"`
- Применяется на текущую ветку
- Ситуация: доделал рабочую часть функциональности

Commit



Commit

...

```
commit 801666fd960d192f857a82cd12393c51f9dde3c4 (HEAD -> main)
Author: Mikhail Kuznetsov <qwerty@gmail.com>
Date:   Tue Jan 10 21:34:46 2023 +0700
```

Commit message

```
diff --git a/swift/SwiftCLI/SwiftCLI/main.swift b/swift/SwiftCLI/SwiftCLI/
main.swift
index fecc4ab..87e90f8 100644
--- a/swift/SwiftCLI/SwiftCLI/main.swift
+++ b/swift/SwiftCLI/SwiftCLI/main.swift
@@ -1 +1,2 @@
-import Foundation
+import UIKit
+print("Hello, Т-Банк!»)
```

Config

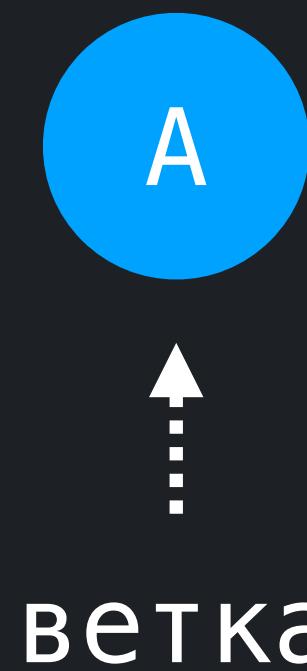
- установка автора и его почты
- `git config user.name USERNAME`
- `git config --global user.email EMAIL`
- Ситуация: ищу того, чей этот коммит

Ветка

- указатель на коммит, уникальное имя
- `git branch NAME`, `git checkout NAME`
- `git branch -d NAME`
- Ситуация: начинаю работать над новой фичей

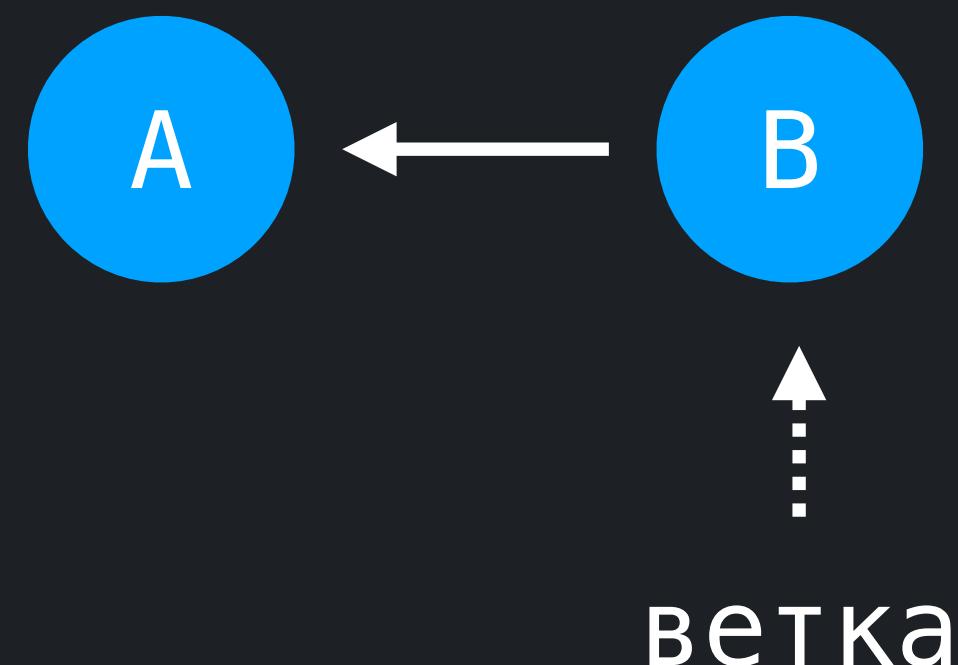
Ветка

- указатель на коммит, уникальное имя
- `git branch NAME`, `git checkout NAME`
- `git branch -d NAME`
- Ситуация: начинаю работать над новой фичей



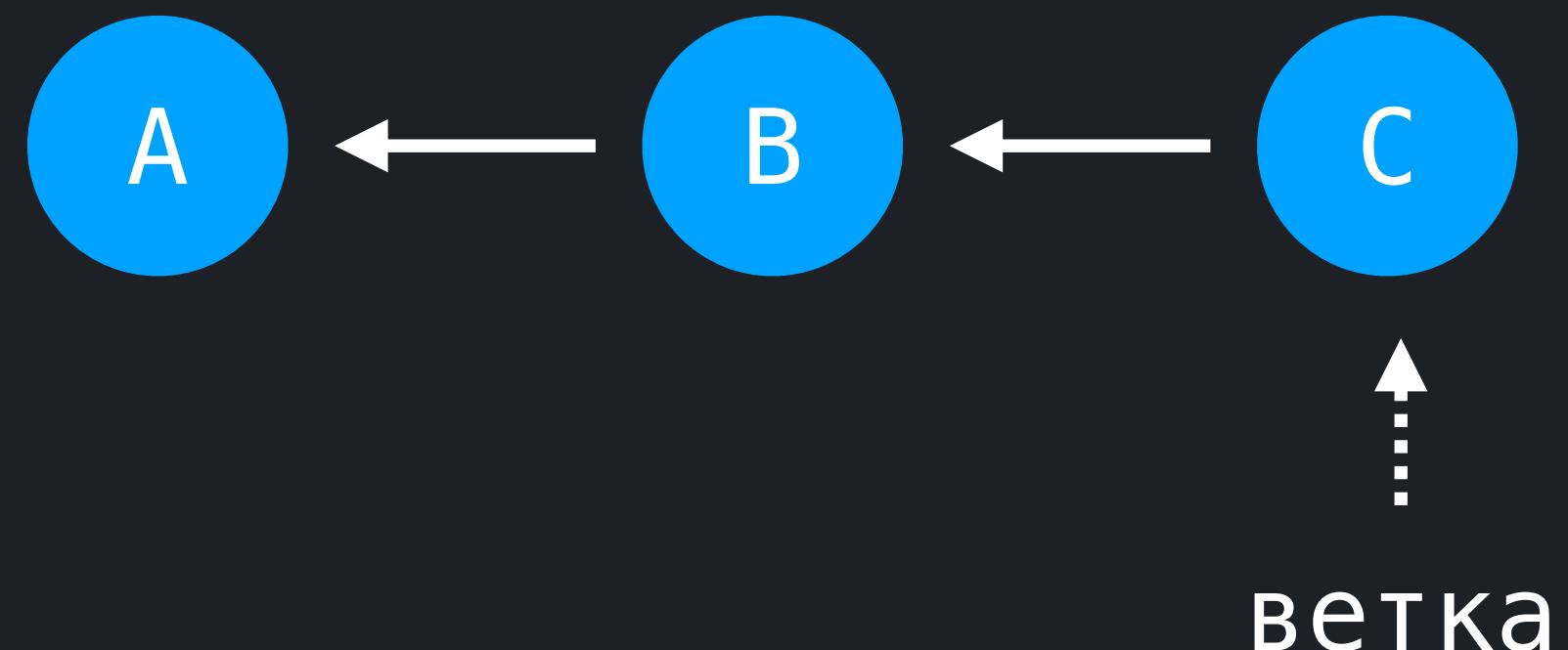
Ветка

- указатель на коммит, уникальное имя
- `git branch NAME`, `git checkout NAME`
- `git branch -d NAME`
- Ситуация: начинаю работать над новой фичей



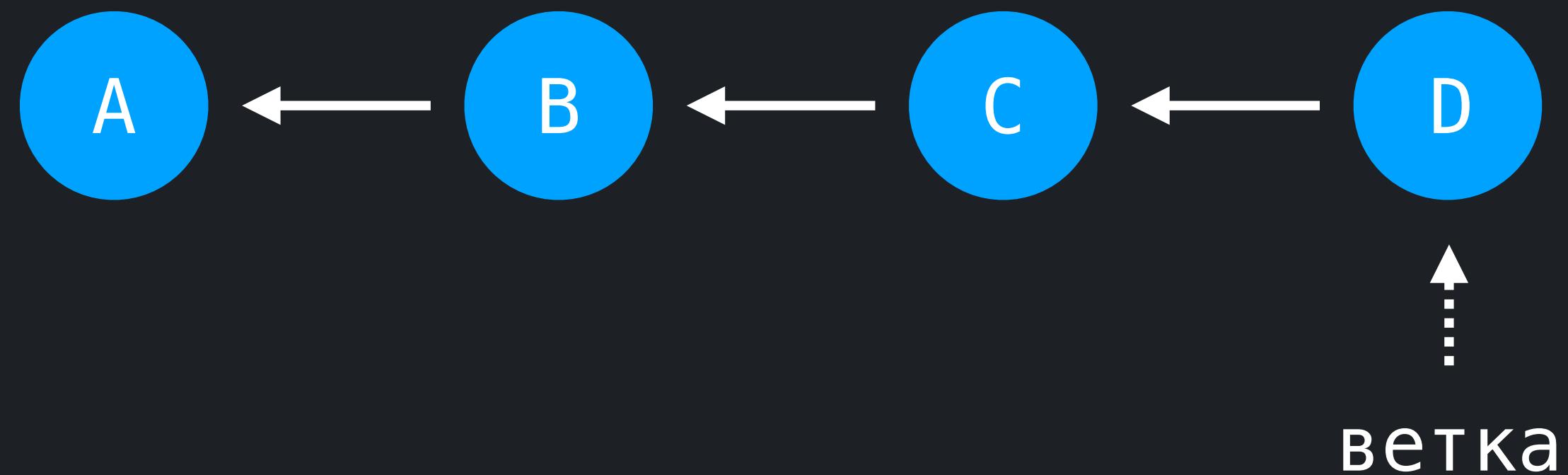
Ветка

- указатель на коммит, уникальное имя
- `git branch NAME`, `git checkout NAME`
- `git branch -d NAME`
- Ситуация: начинаю работать над новой фичей



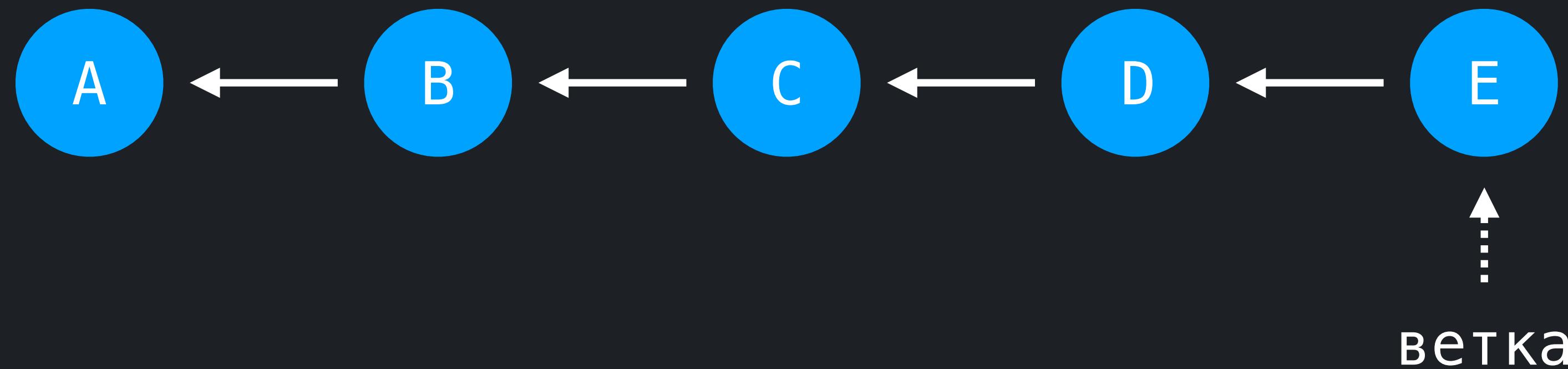
Ветка

- указатель на коммит, уникальное имя
- `git branch NAME`, `git checkout NAME`
- `git branch -d NAME`
- Ситуация: начинаю работать над новой фичей



Ветка

- указатель на коммит, уникальное имя
- `git branch NAME`, `git checkout NAME`
- `git branch -d NAME`
- Ситуация: начинаю работать над новой фичей



Тэг

- указатель на коммит
- имеет уникальное имя
- зафиксирован
- `git tag NAME`
- Ситуация: хочу запомнить версию, которую отправили в AppStore

История

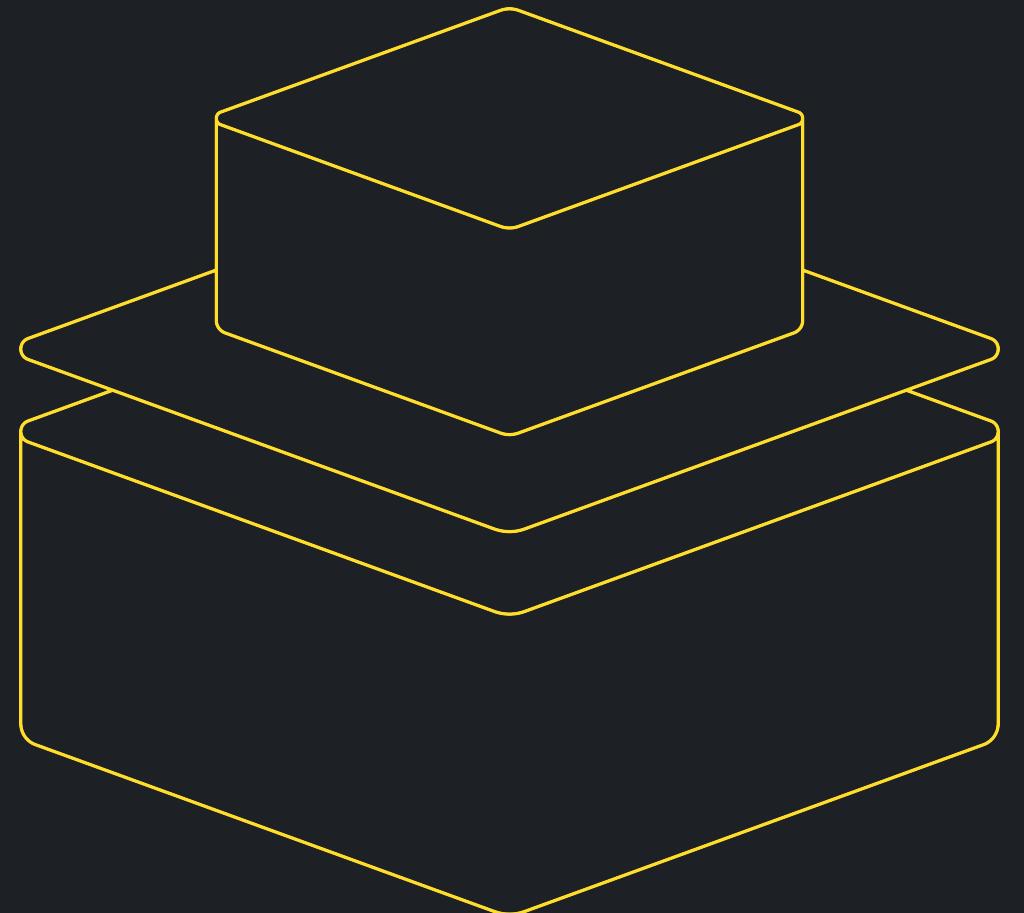
- `git log SHA` просмотр коммита
- `git log -p BRANCH` детальный просмотр
- GUI нагляднее
- Ситуация: хочу посмотреть что есть в проекте

Remote

- удаленный компьютер (чаще всего)
- **точно** такой же
- `git remote add NAME URL`
- Ситуация: подключаю общий репозиторий с кодом

Способы подключения

- ✓ ssh: keys
- ! https: username + password



Tracking

- установка отслеживания указателей
- `git branch -u REMOTE/BRANCH`
- Ситуация: создаю новый экран в приложении

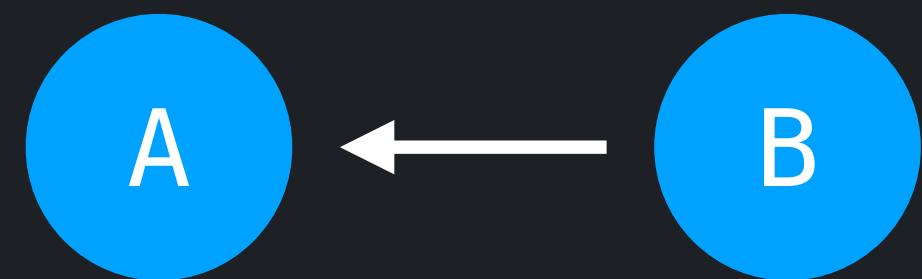
Tracking

- установка отслеживания указателей
- `git branch -u REMOTE/BRANCH`
- Ситуация: создаю новый экран в приложении

A

Tracking

- установка отслеживания указателей
- `git branch -u REMOTE/BRANCH`
- Ситуация: создаю новый экран в приложении



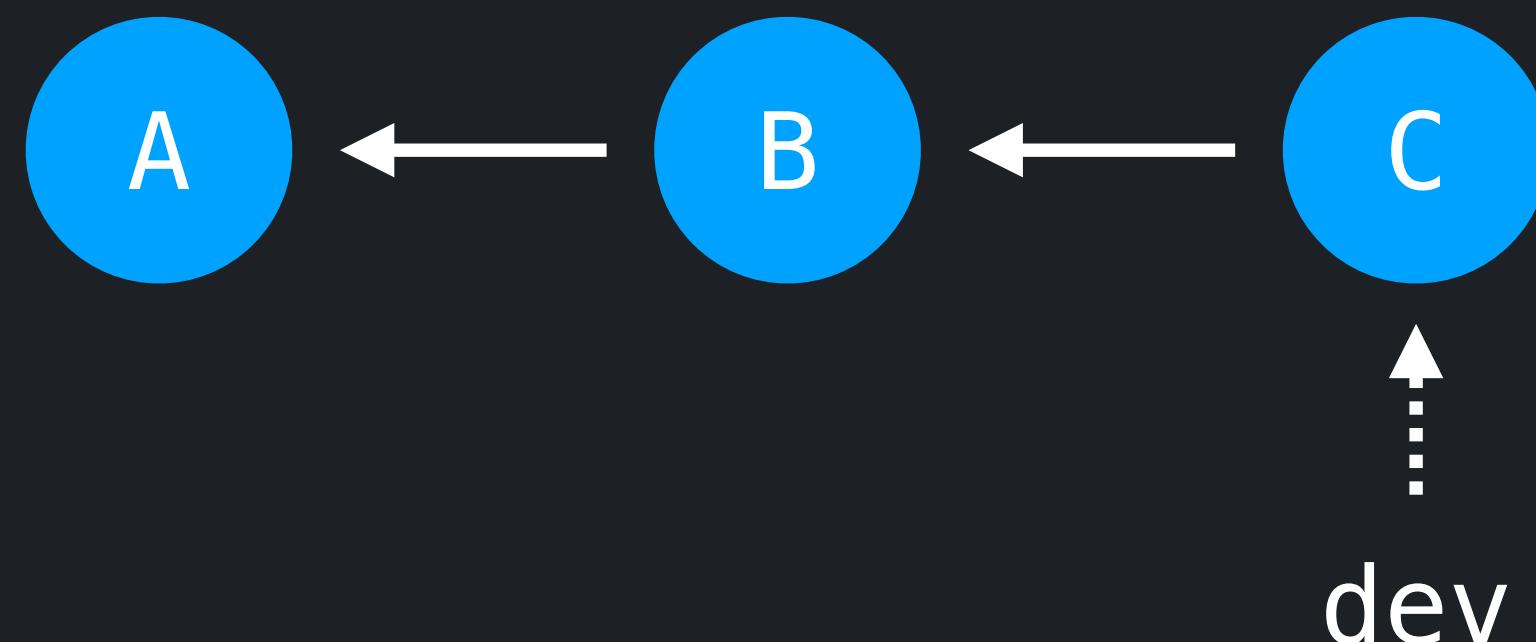
Tracking

- установка отслеживания указателей
- `git branch -u REMOTE/BRANCH`
- Ситуация: создаю новый экран в приложении



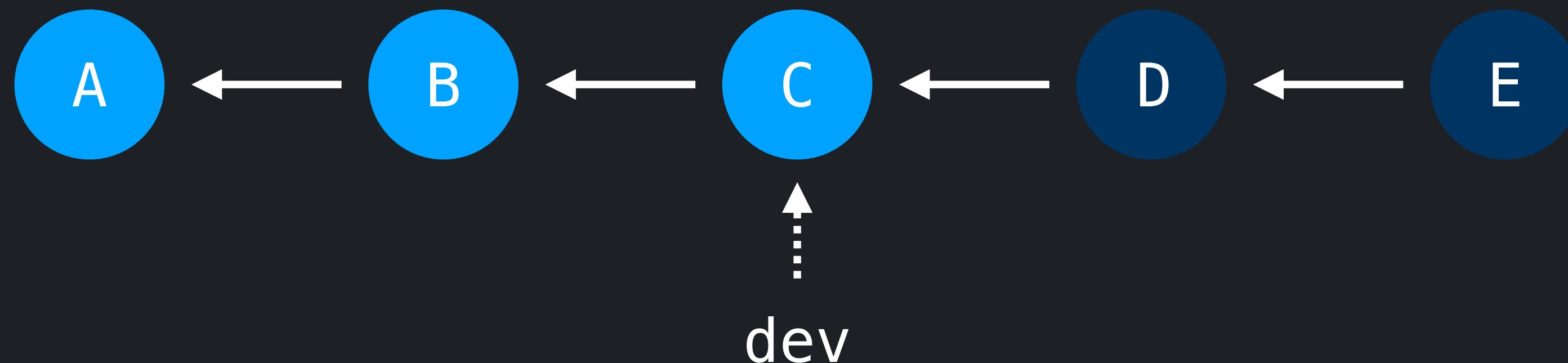
Tracking

- установка отслеживания указателей
- `git branch -u REMOTE/BRANCH`
- Ситуация: создаю новый экран в приложении



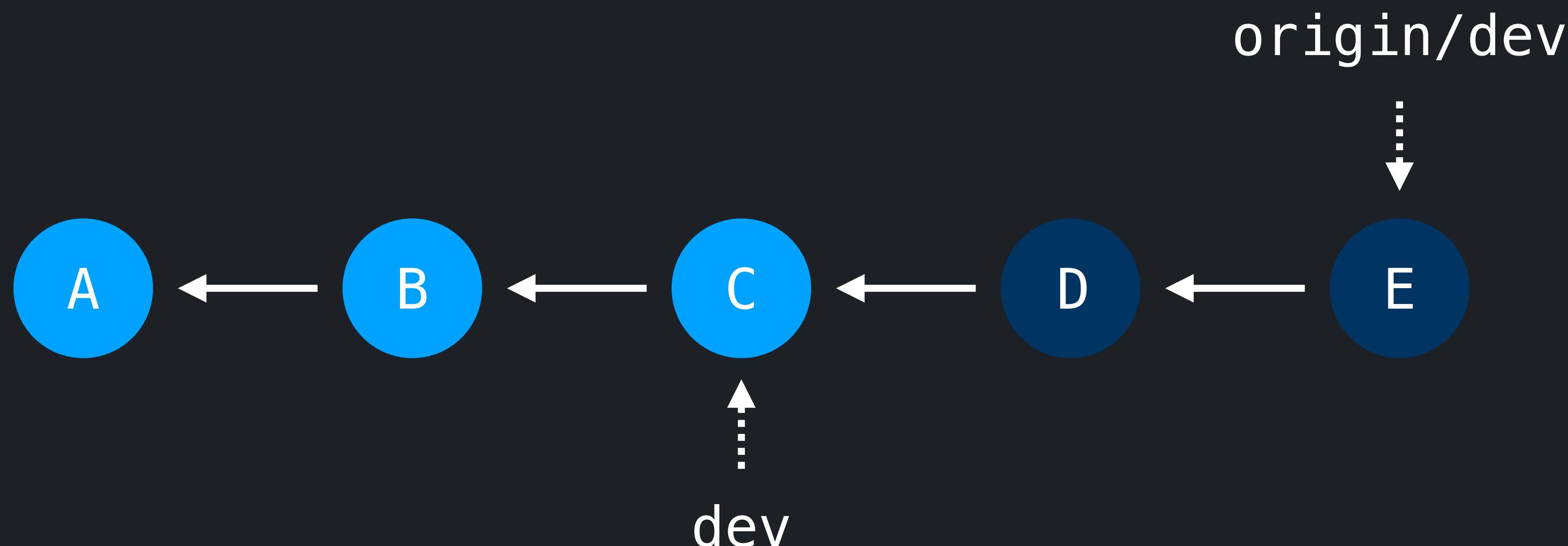
Tracking

- установка отслеживания указателей
- `git branch -u REMOTE/BRANCH`
- Ситуация: создаю новый экран в приложении



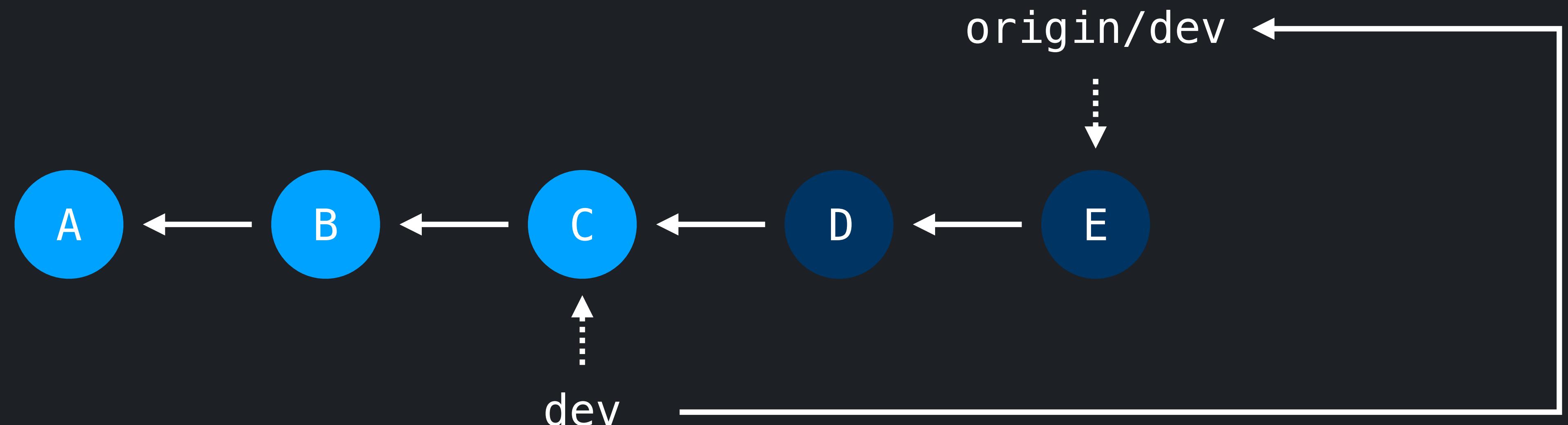
Tracking

- установка отслеживания указателей
- `git branch -u REMOTE/BRANCH`
- Ситуация: создаю новый экран в приложении



Tracking

- установка отслеживания указателей
- `git branch -u REMOTE/BRANCH`
- Ситуация: создаю новый экран в приложении



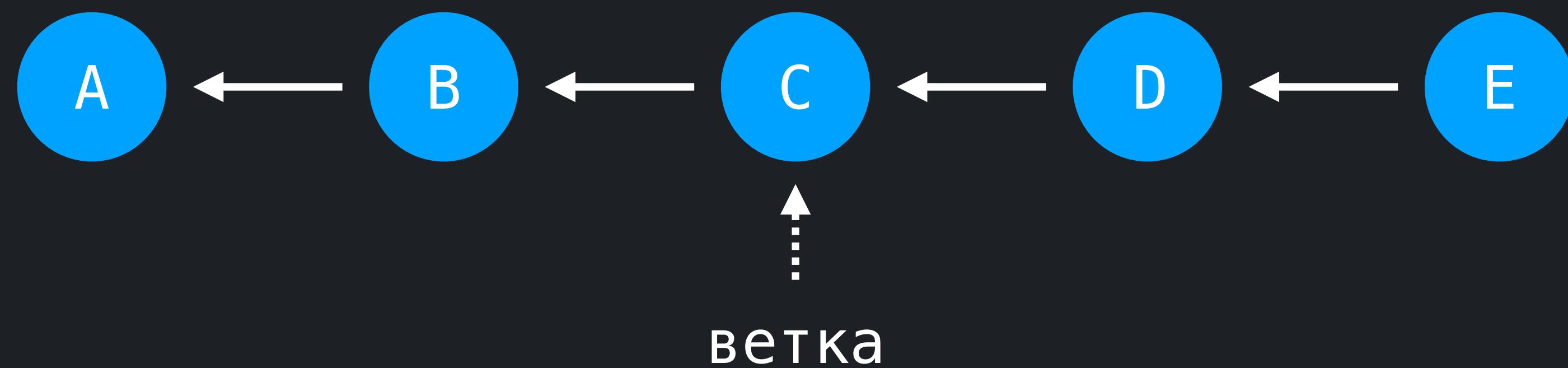
Fast-Forward

- процесс перемещения указателя вперед



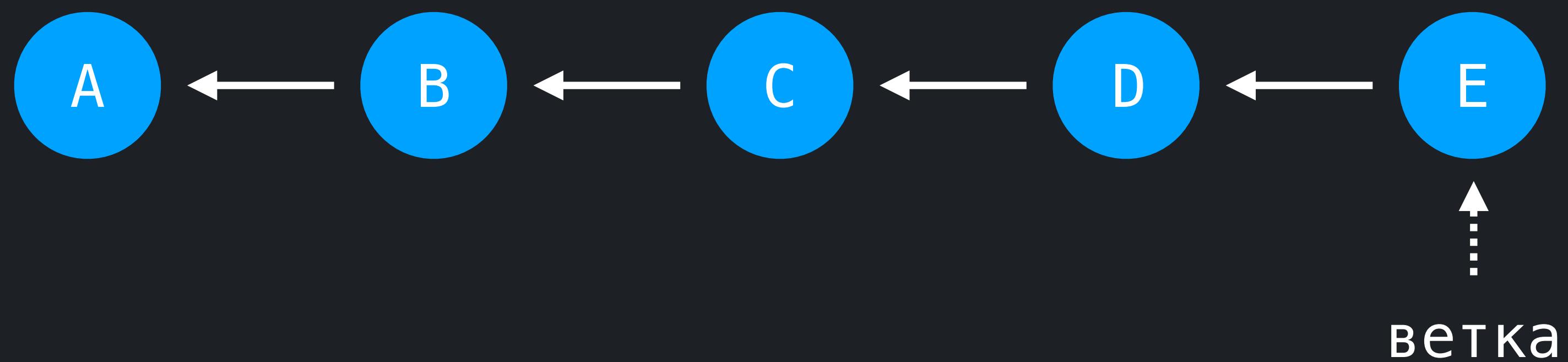
Fast-Forward

- процесс перемещения указателя вперед



Fast-Forward

- процесс перемещения указателя вперед



Push

- отправляет коммиты, меняет указатели
- `git push origin dev`
- Ситуация: отправляю фичу на CI

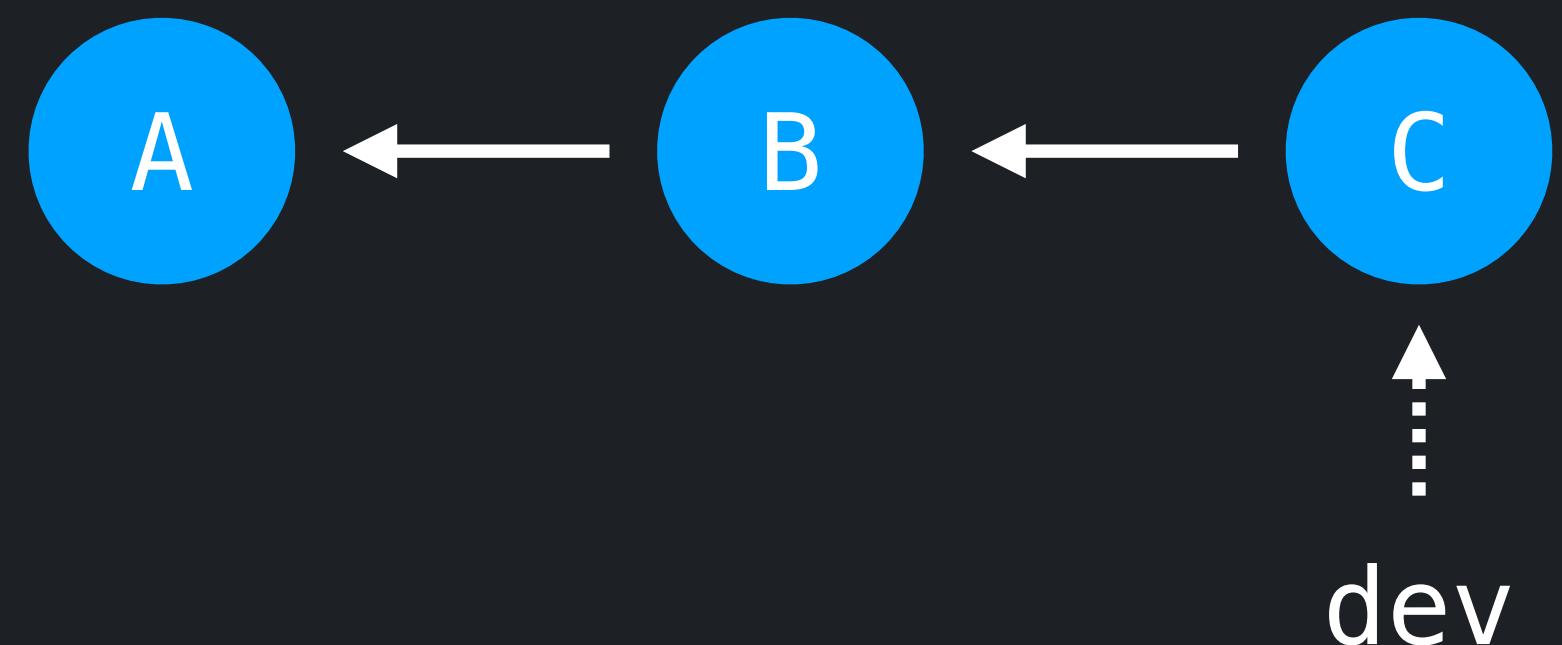
Push

- отправляет коммиты, меняет указатели
- `git push origin dev`
- Ситуация: отправляю фичу на CI



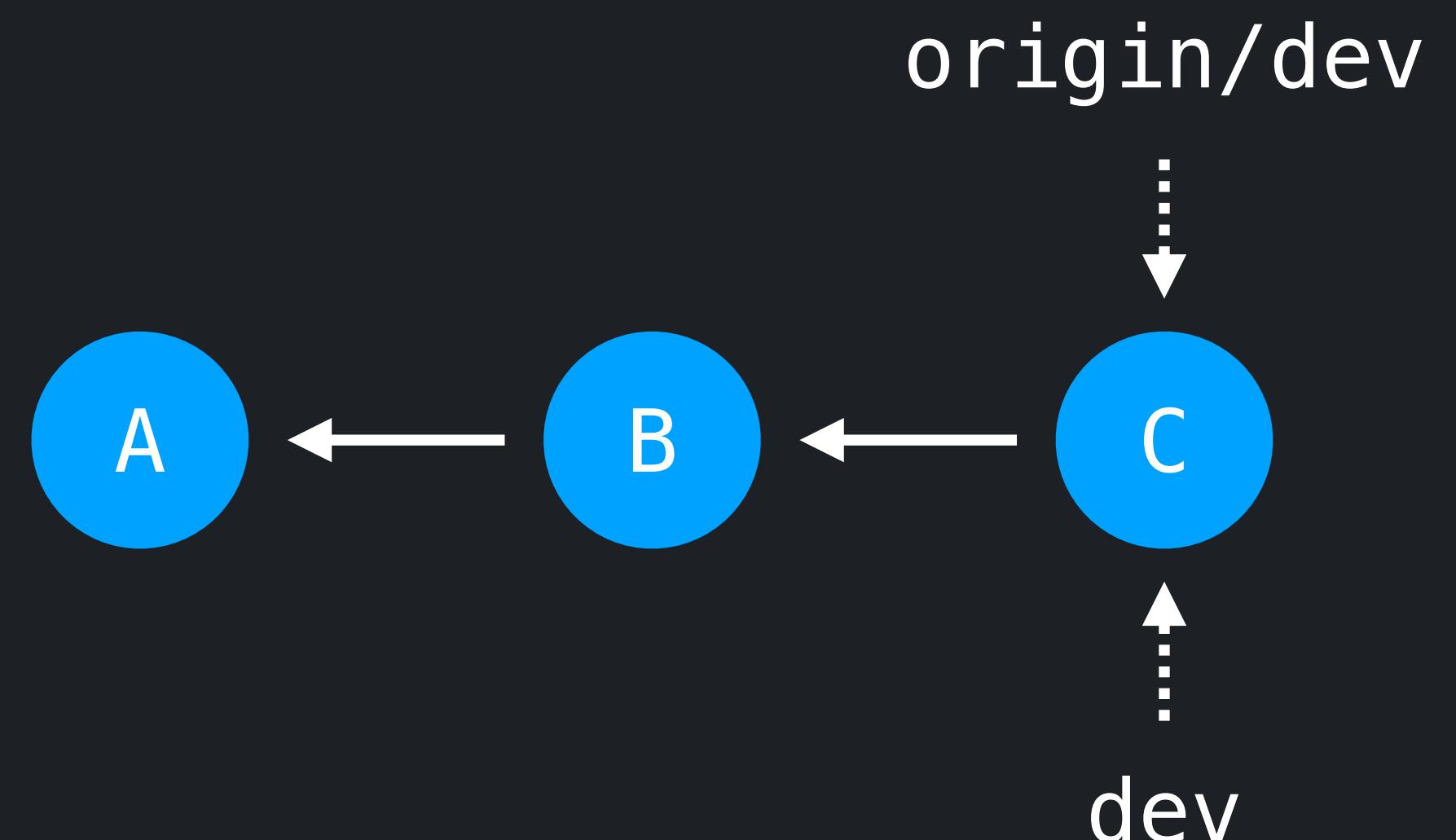
Push

- отправляет коммиты, меняет указатели
- `git push origin dev`
- Ситуация: отправляю фичу на CI



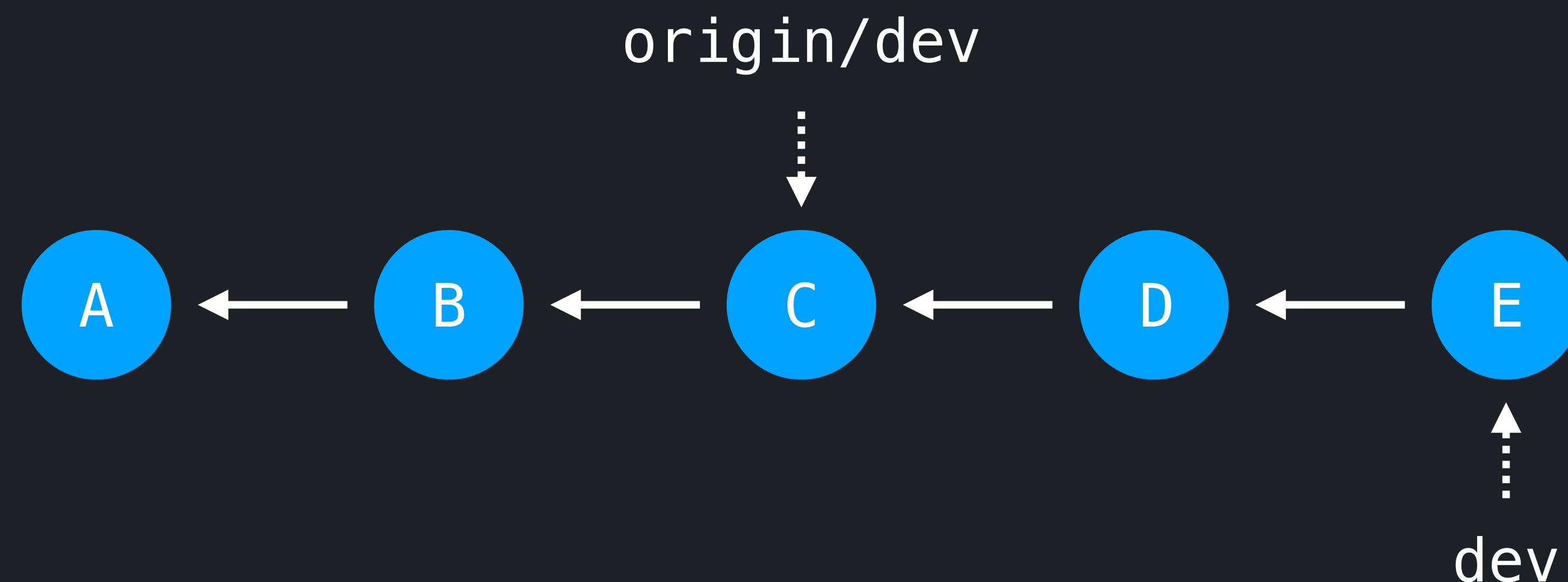
Push

- отправляет коммиты, меняет указатели
- `git push origin dev`
- Ситуация: отправляю фичу на CI



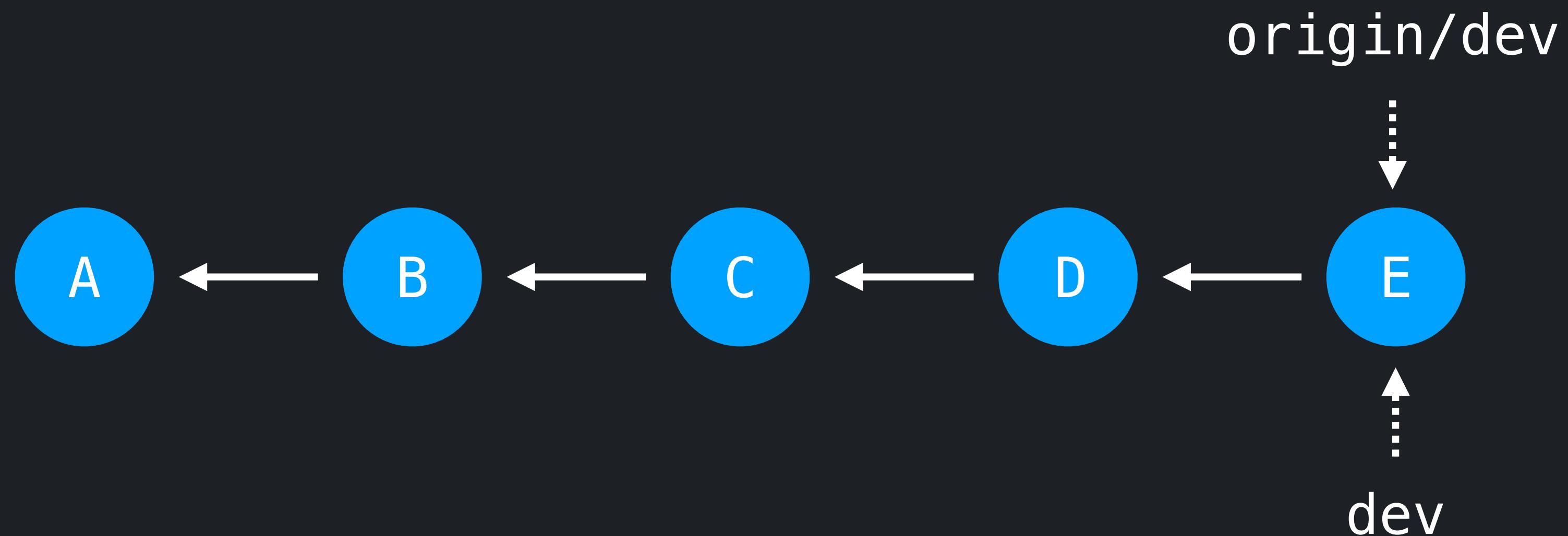
Push

- отправляет коммиты, меняет указатели
- `git push origin dev`
- Ситуация: отправляю фичу на CI



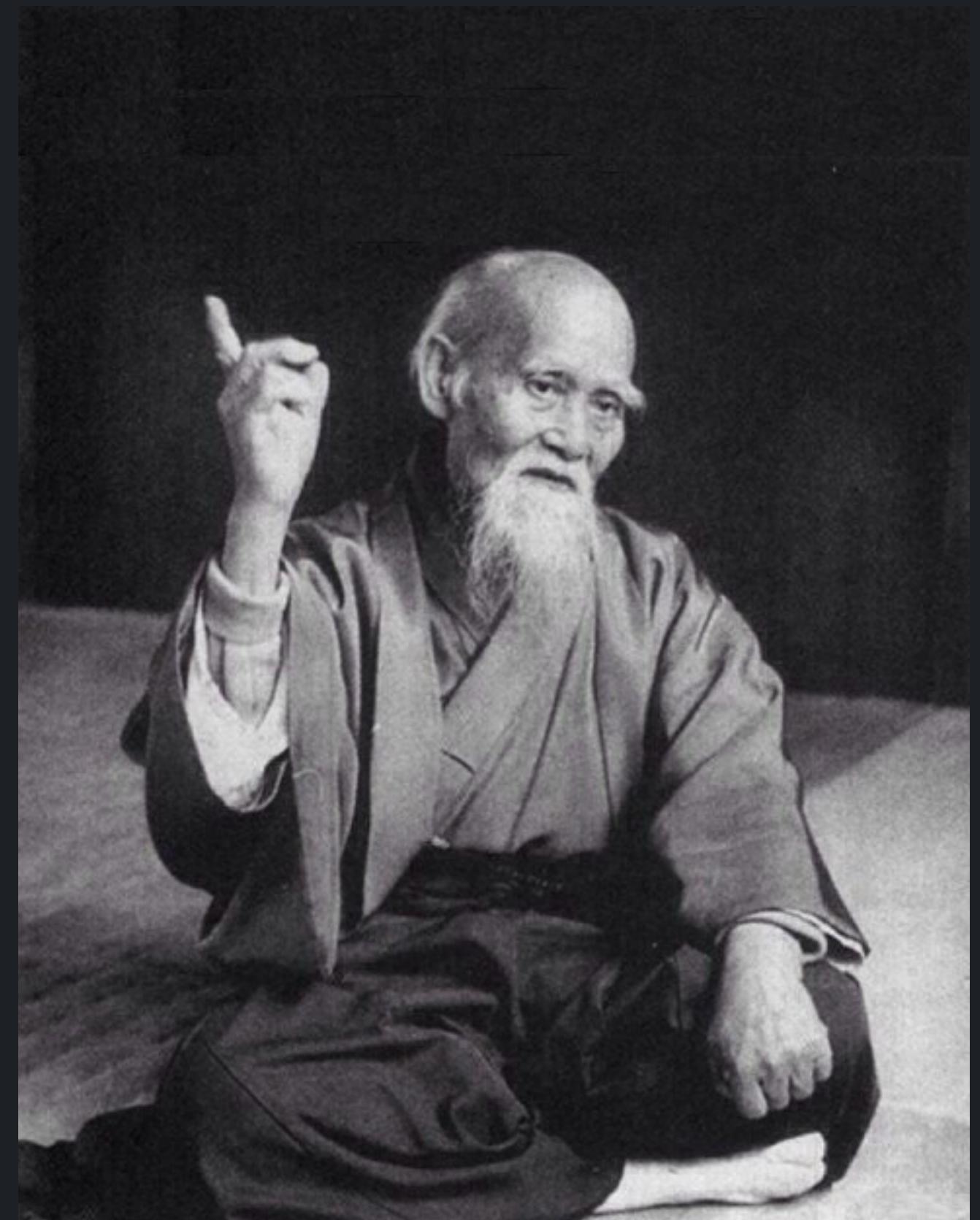
Push

- отправляет коммиты, меняет указатели
- `git push origin dev`
- Ситуация: отправляю фичу на CI



Важно

- КОММИТ = атомарные изменения
- используйте осмысленные именования,
feature/JIRA-XXX-awesome-feature
- следуйте соглашению в компании/
команде
- делайте заметки через Stash



Terminal vs GUI

- разницы нет, так как знание основ **первостепенно**
- GUI лишь сокращает время на рутинные операции
- в повседневной жизни пользуюсь GUI

Основные команды

Fetch

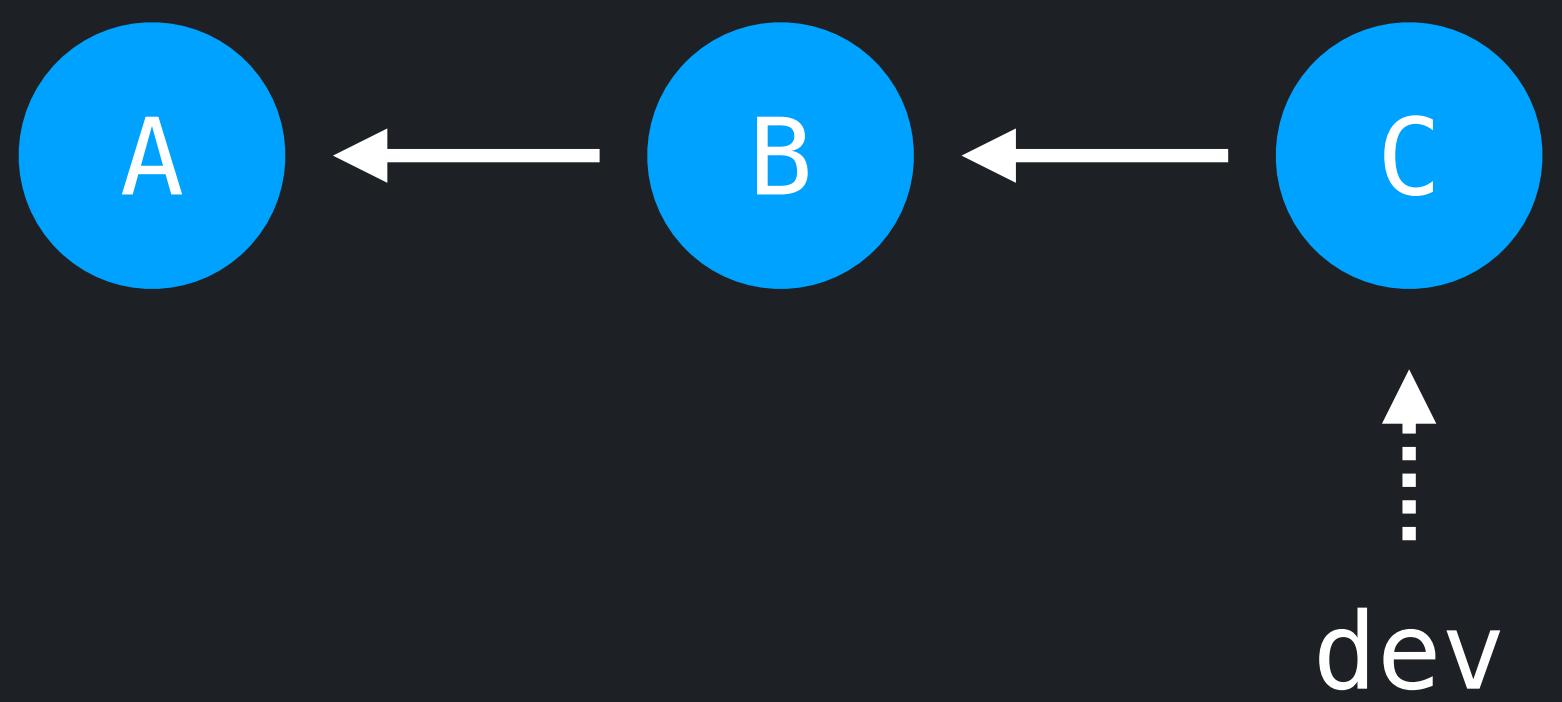
- `git fetch origin BRANCH`
- загружает изменения
- не меняет локальных указателей (ветки)
- Ситуация: узнаю что за изменения на remote

Fetch

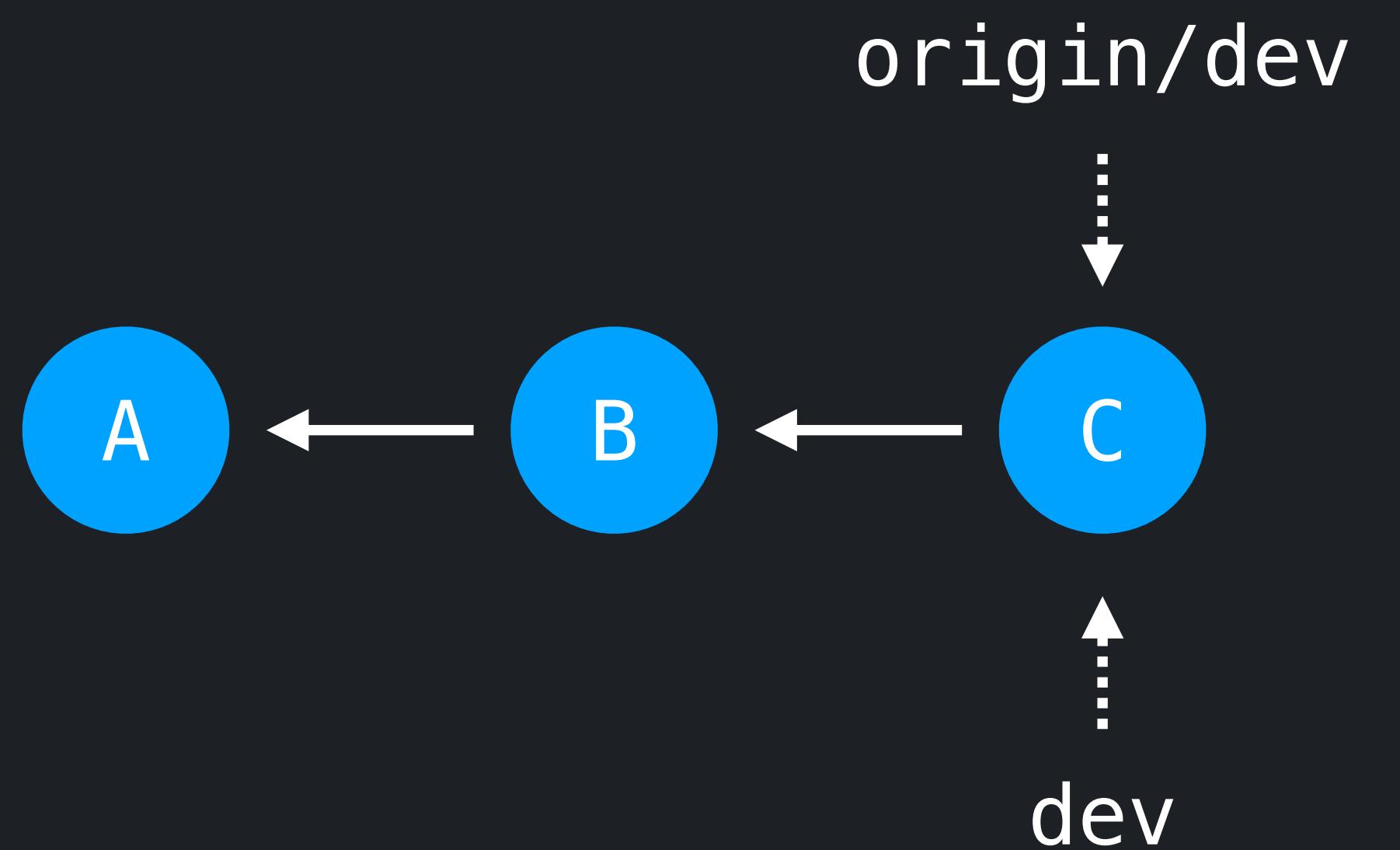
Fetch



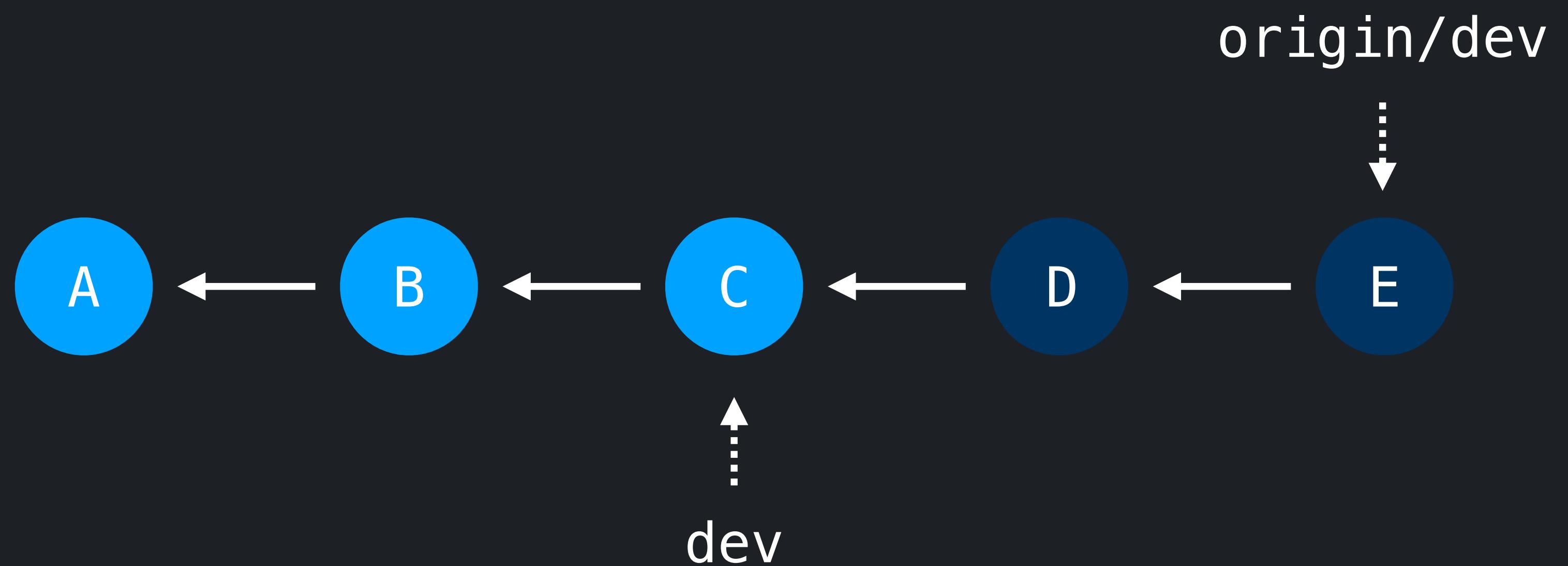
Fetch



Fetch



Fetch



Сброс коммита

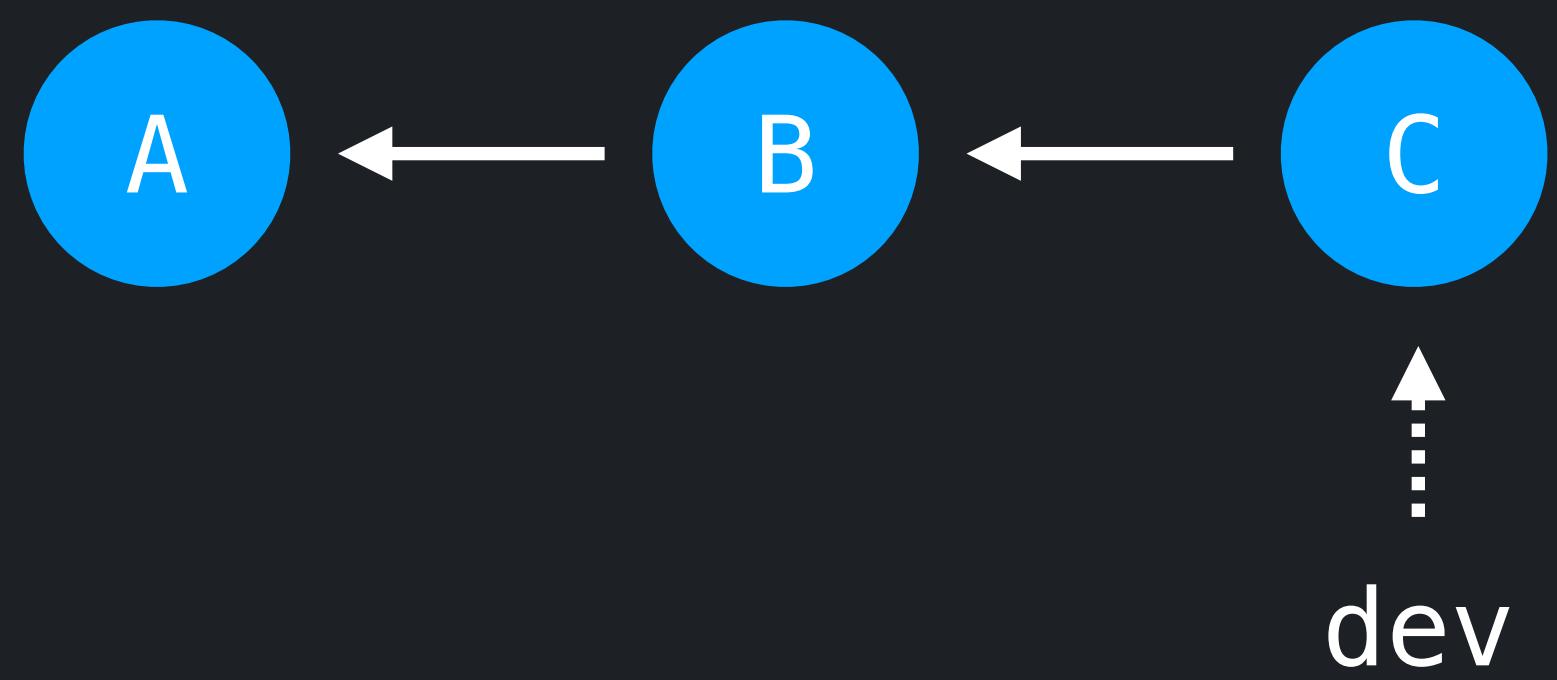
- `git reset COMMIT`
- три основных режима
 - `--soft` файлы не меняются, готовы к коммиту
 - `--mixed` файлы не меняются, не готовы к коммиту
 - `--hard` файлы сбрасывают до предыдущего состояния
- Ситуация: хочу заменить коммит

Сброс коммита

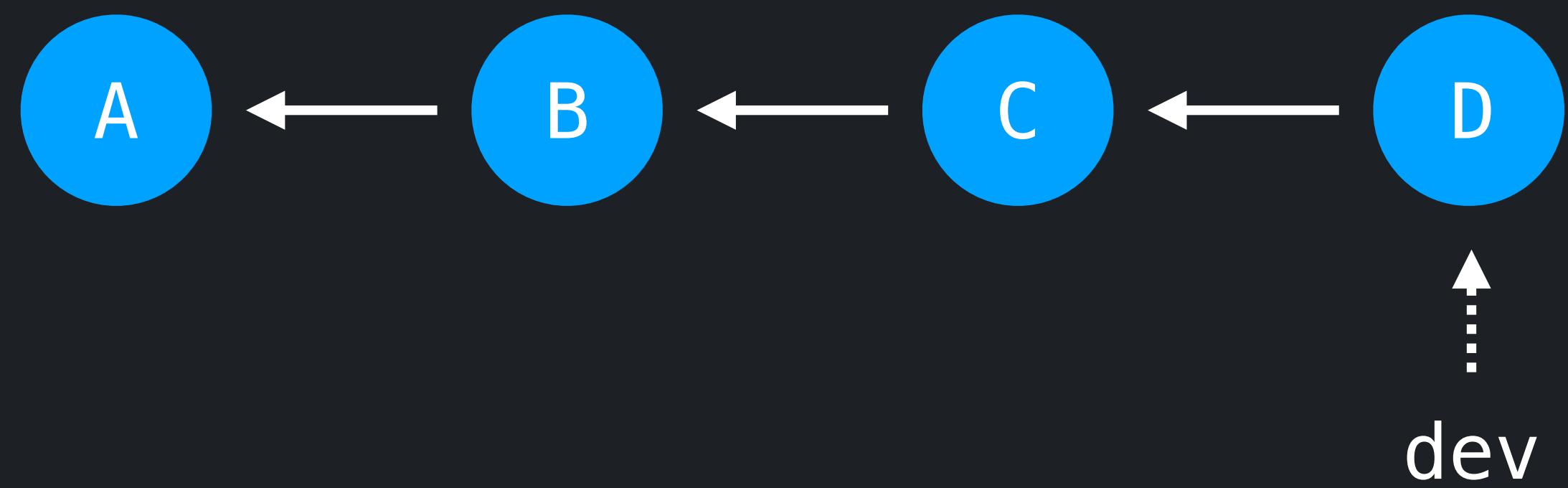
Сброс коммита



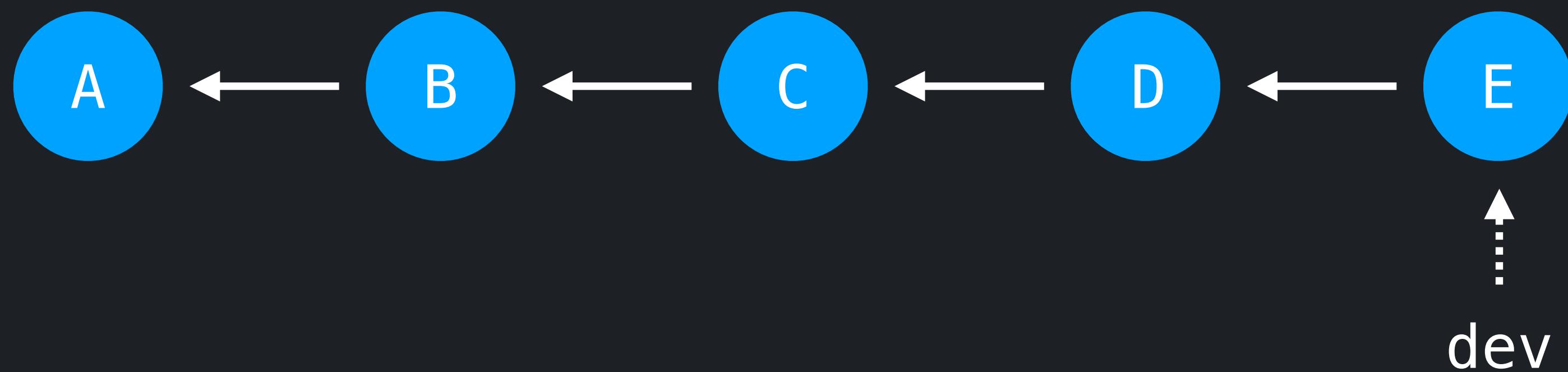
Сброс коммита



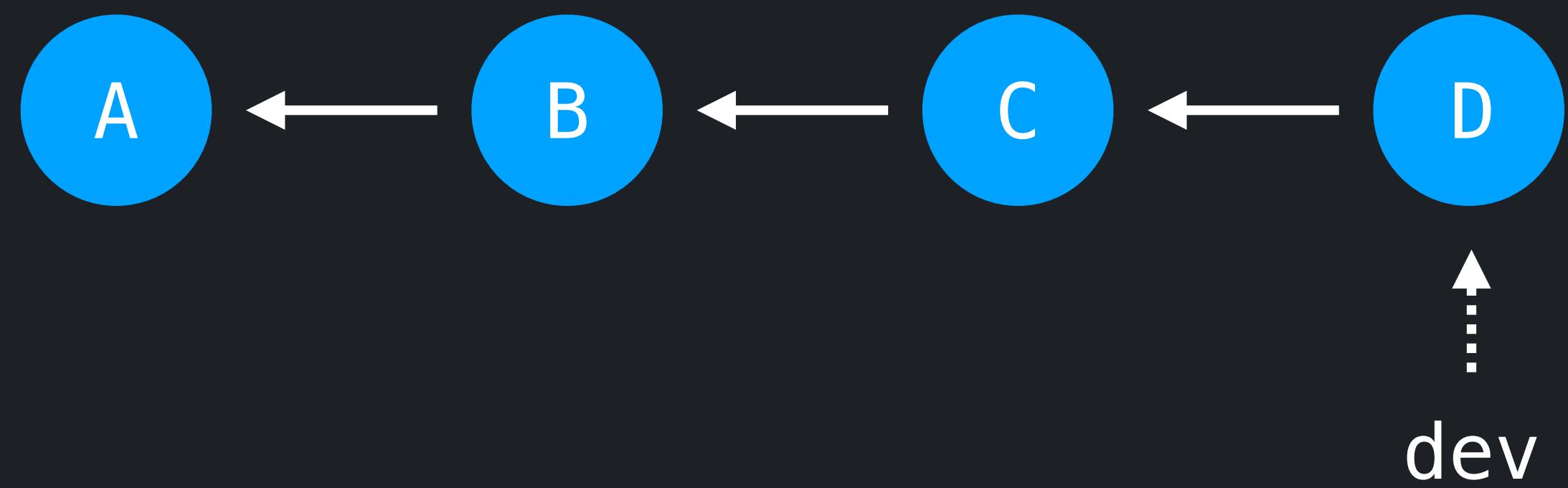
Сброс коммита



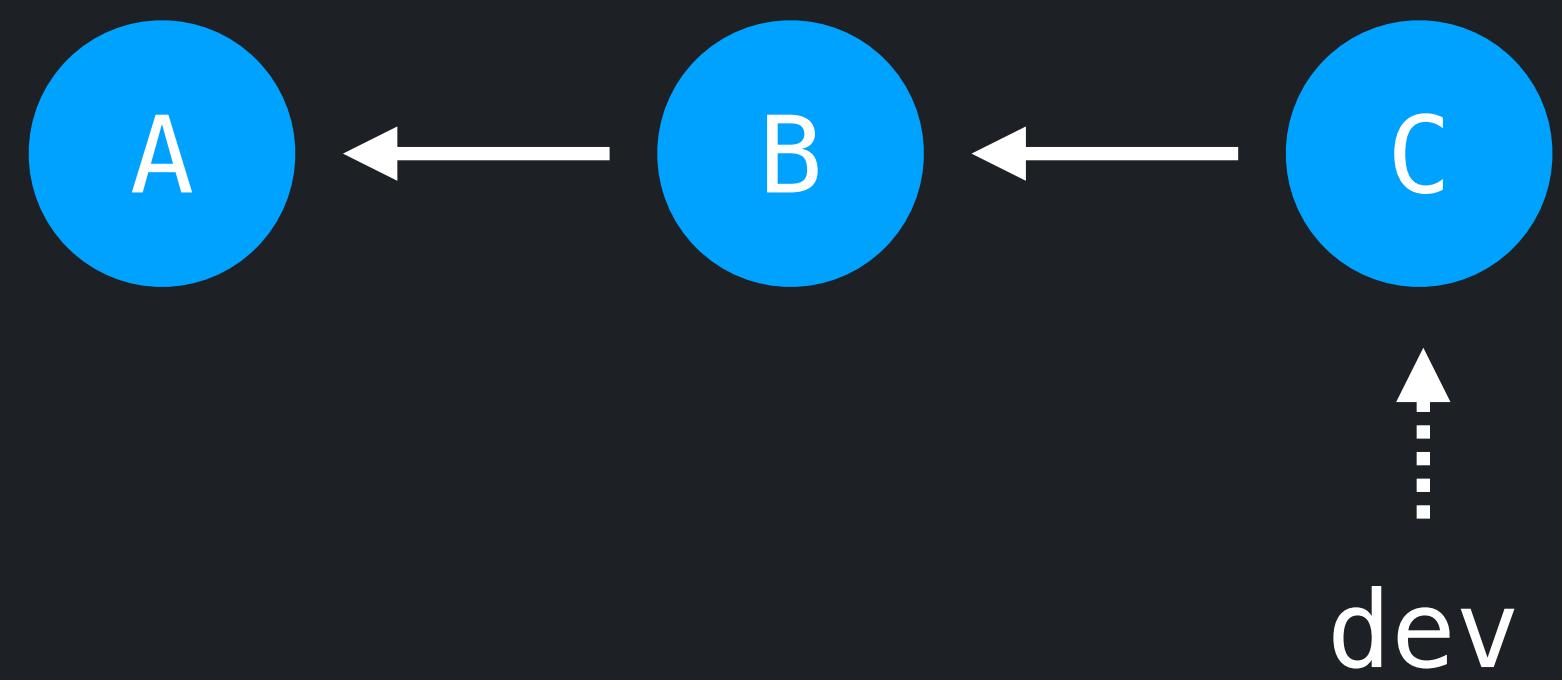
Сброс коммита



Сброс коммита

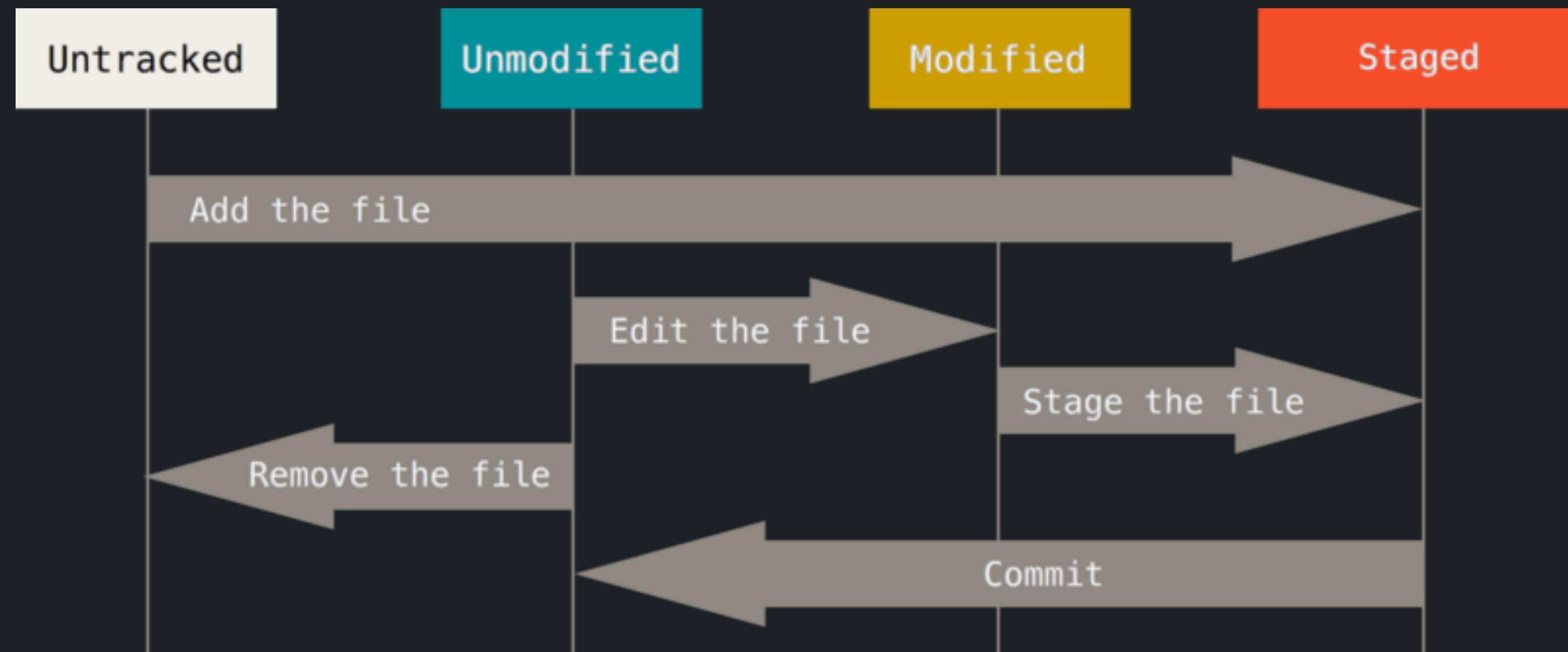


Сброс коммита



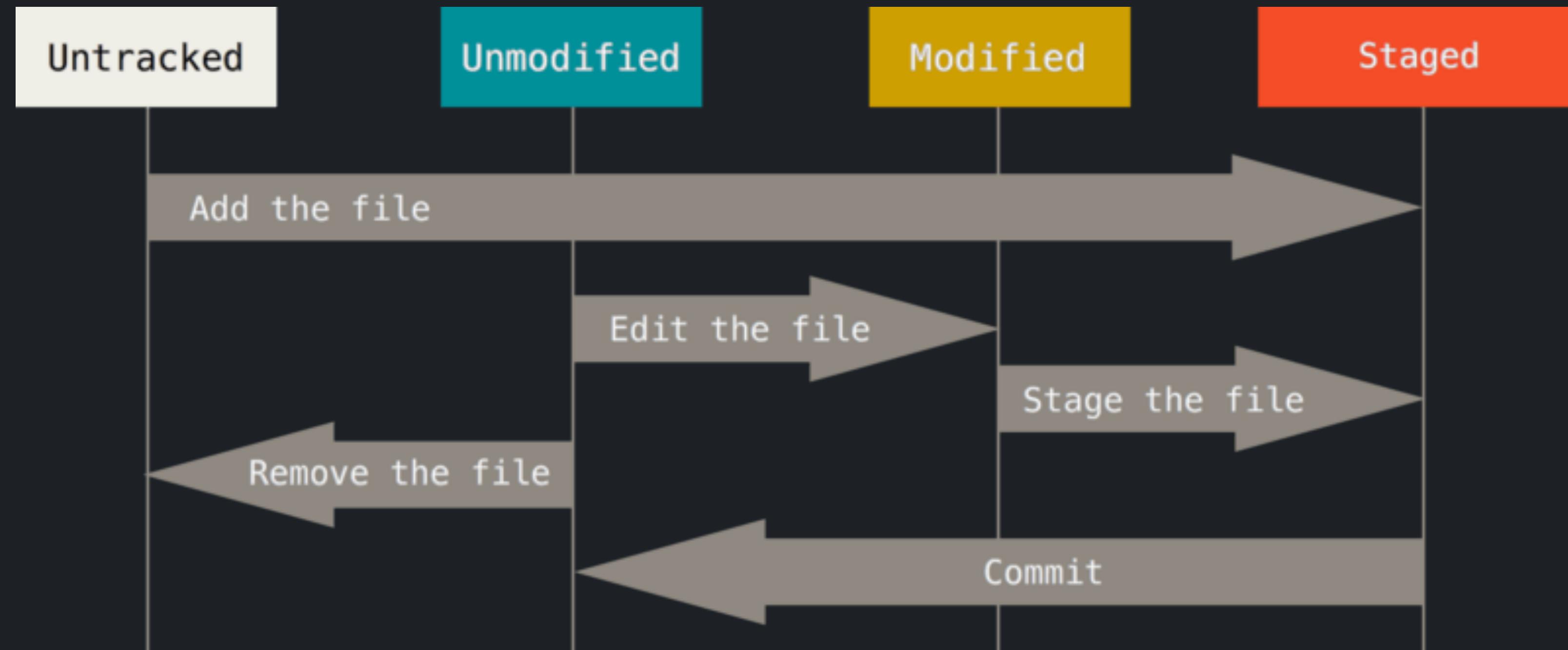
Состояния после сброса

Состояния после сброса

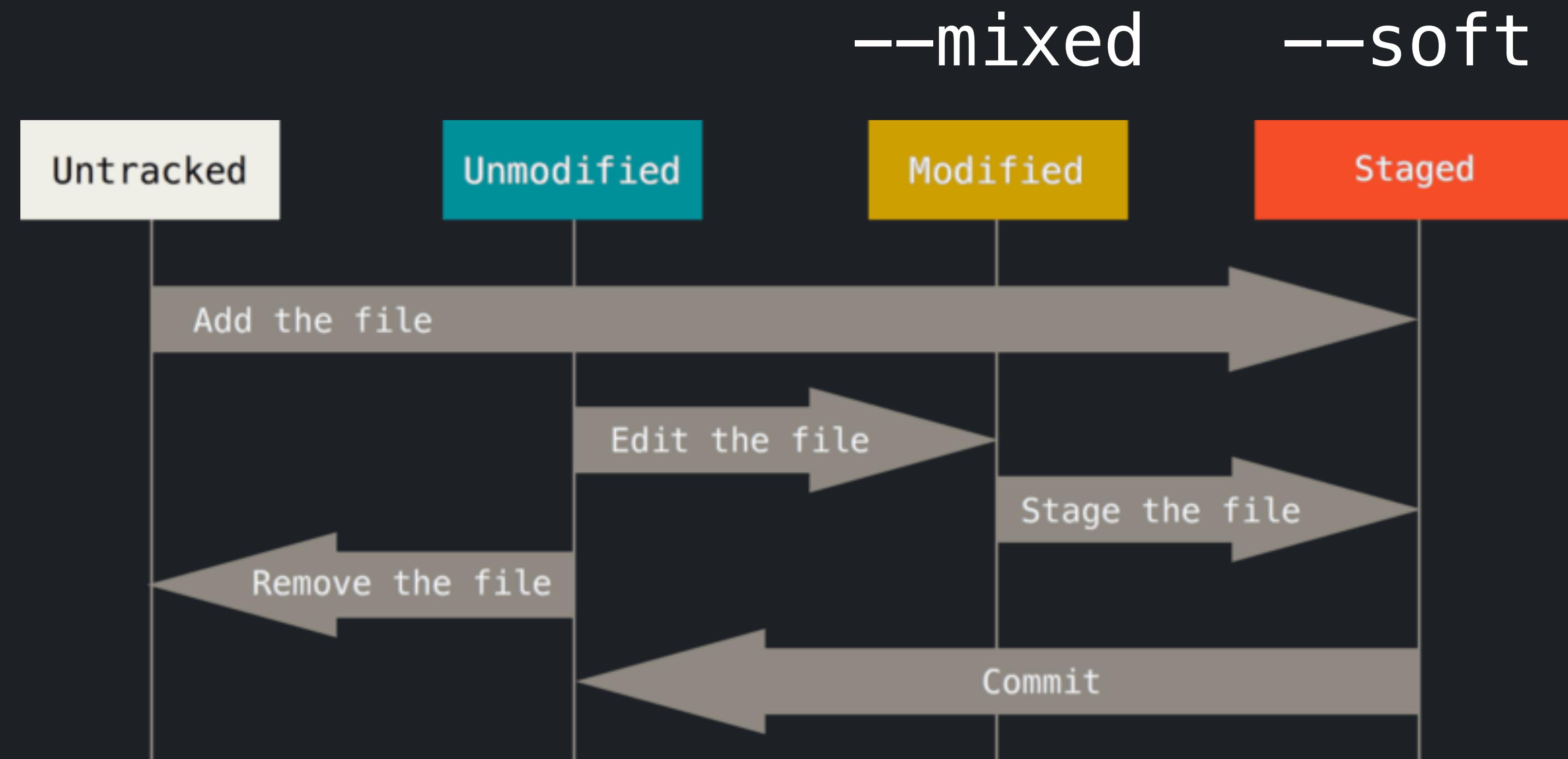


Состояния после сброса

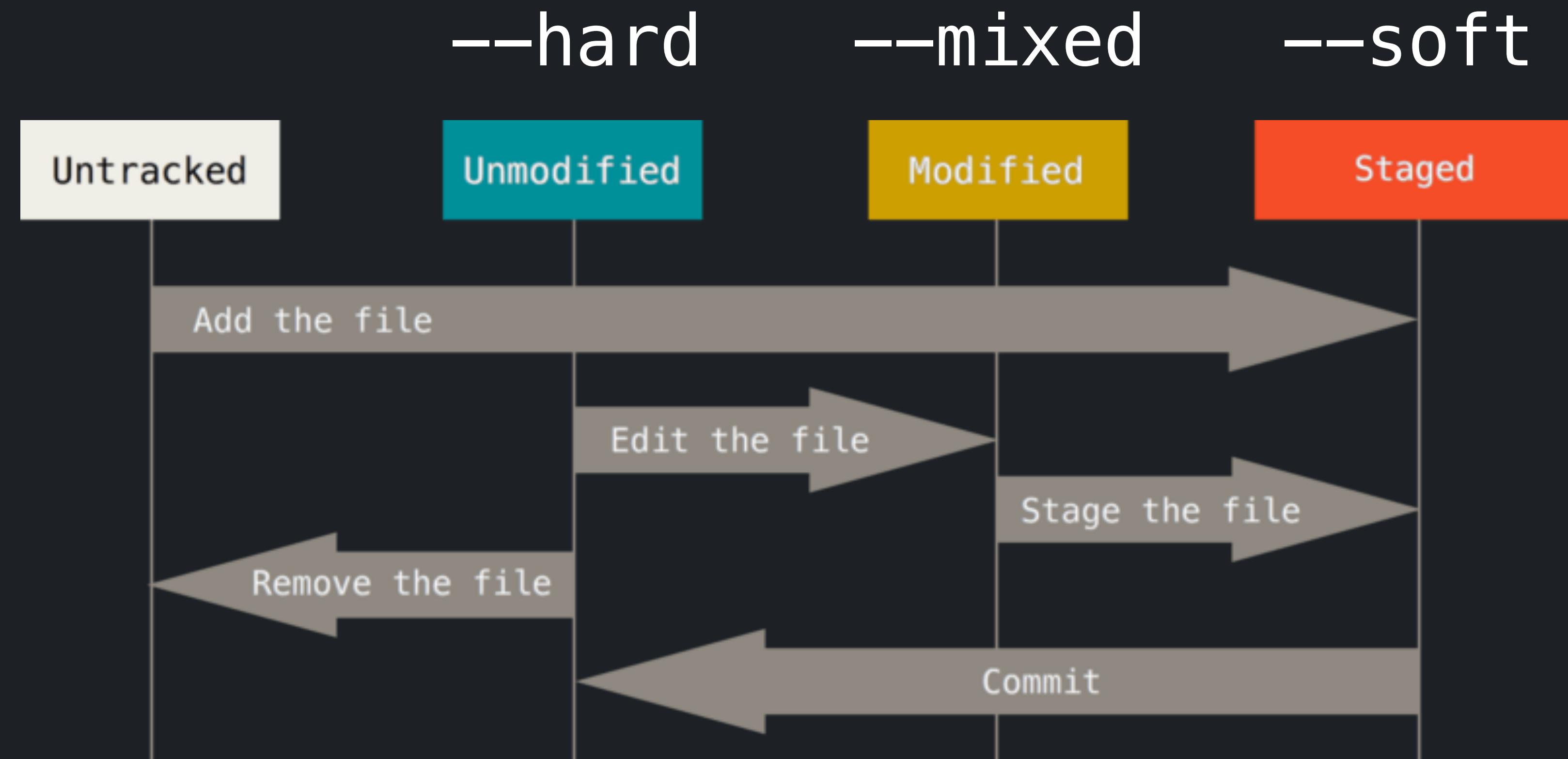
--soft



Состояния после сброса



Состояния после сброса



Amend (squash)

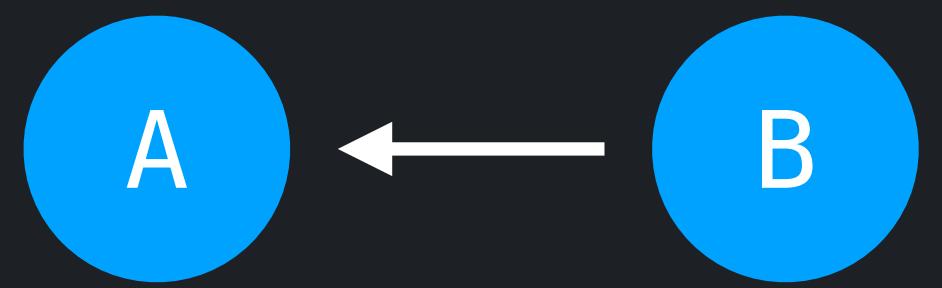
- соединяет коммит с родителем
- `git commit --amend` = `git reset --soft && git commit`
- Ситуация: хочу дозаписать в прошлый коммит

Merge

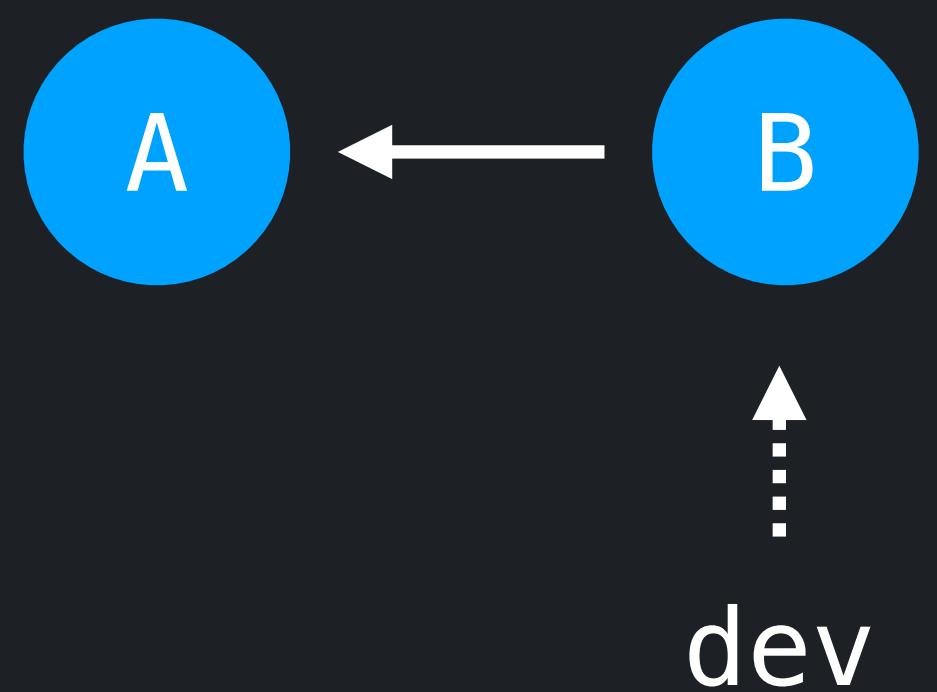
- добавляет изменения с другой ветки
- создает еще один коммит
- `git merge BRANCH`
- Ситуация: хочу соединить изменения с двух веток

Merge

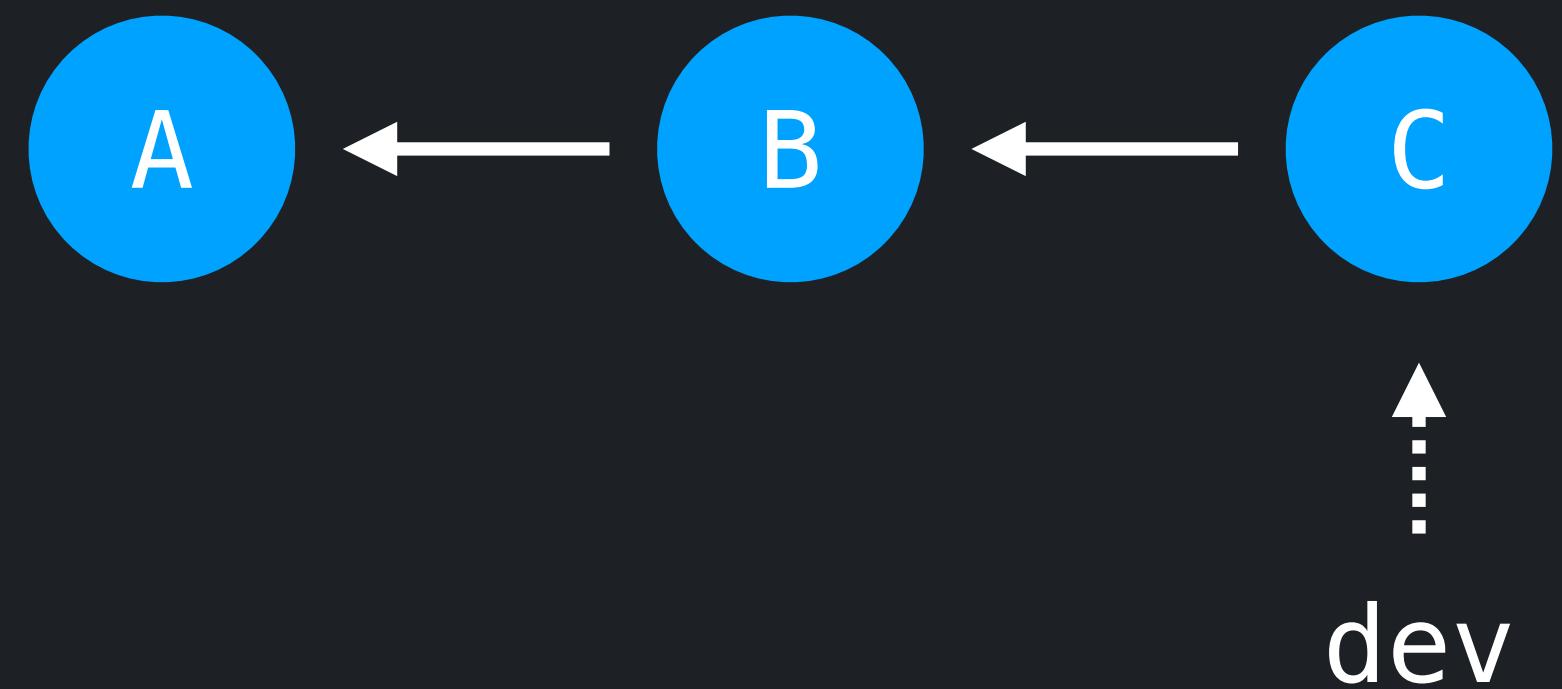
Merge



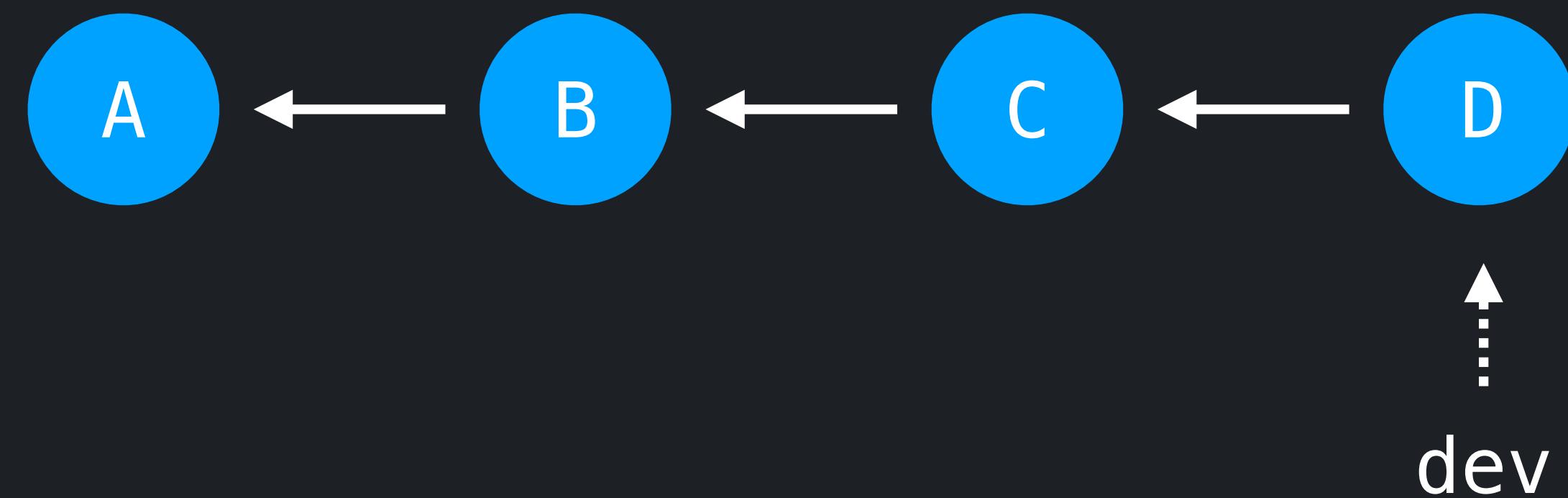
Merge



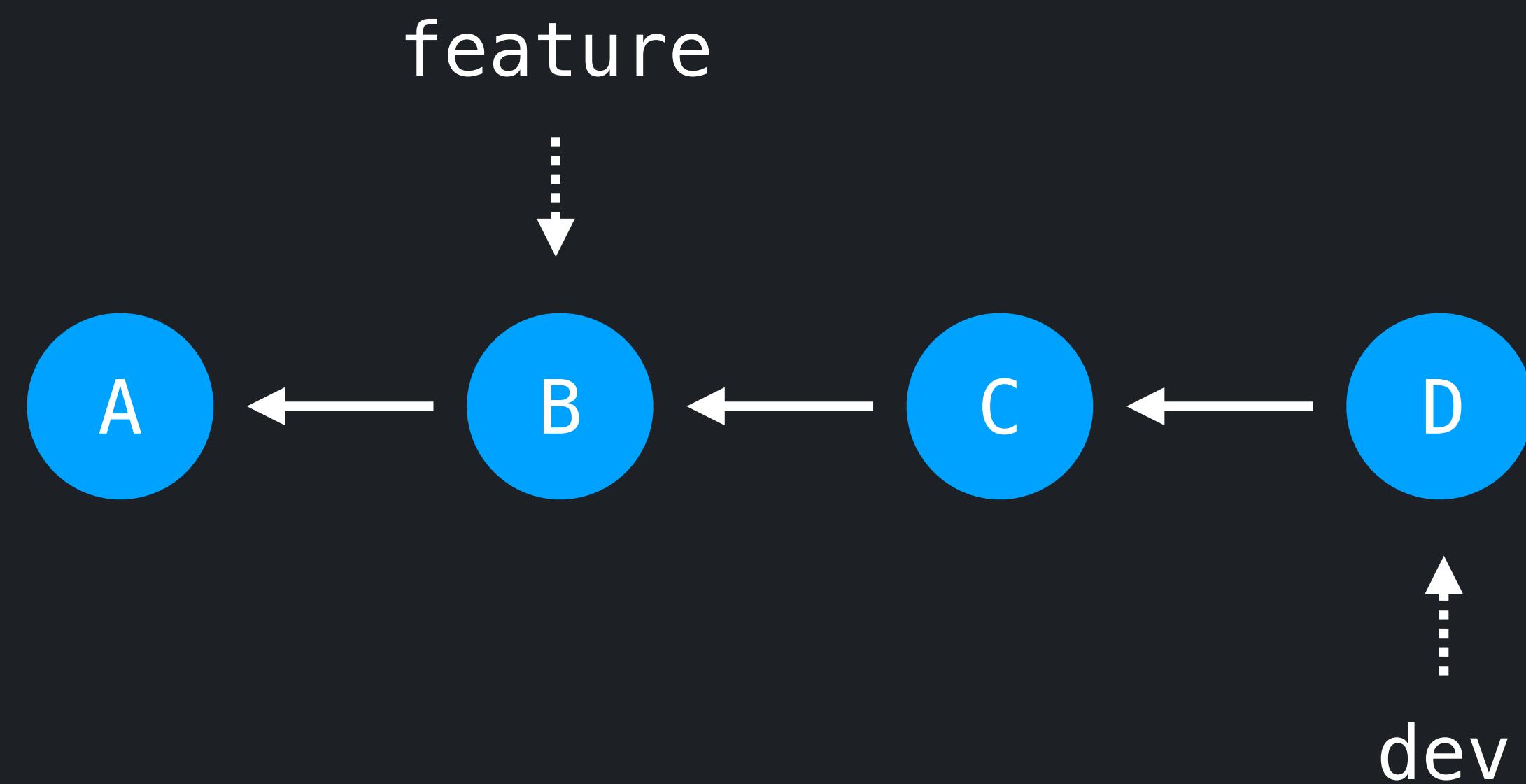
Merge



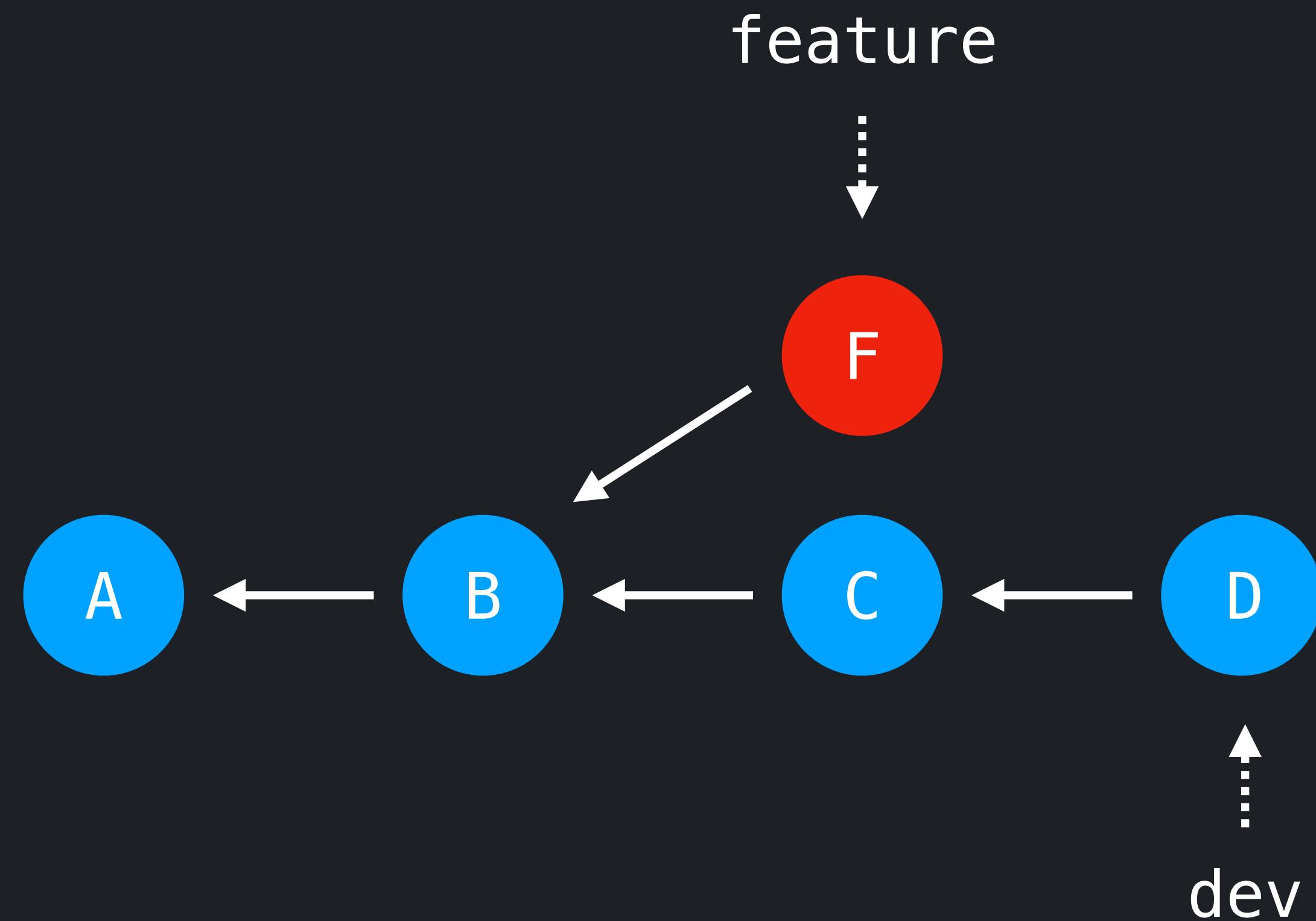
Merge



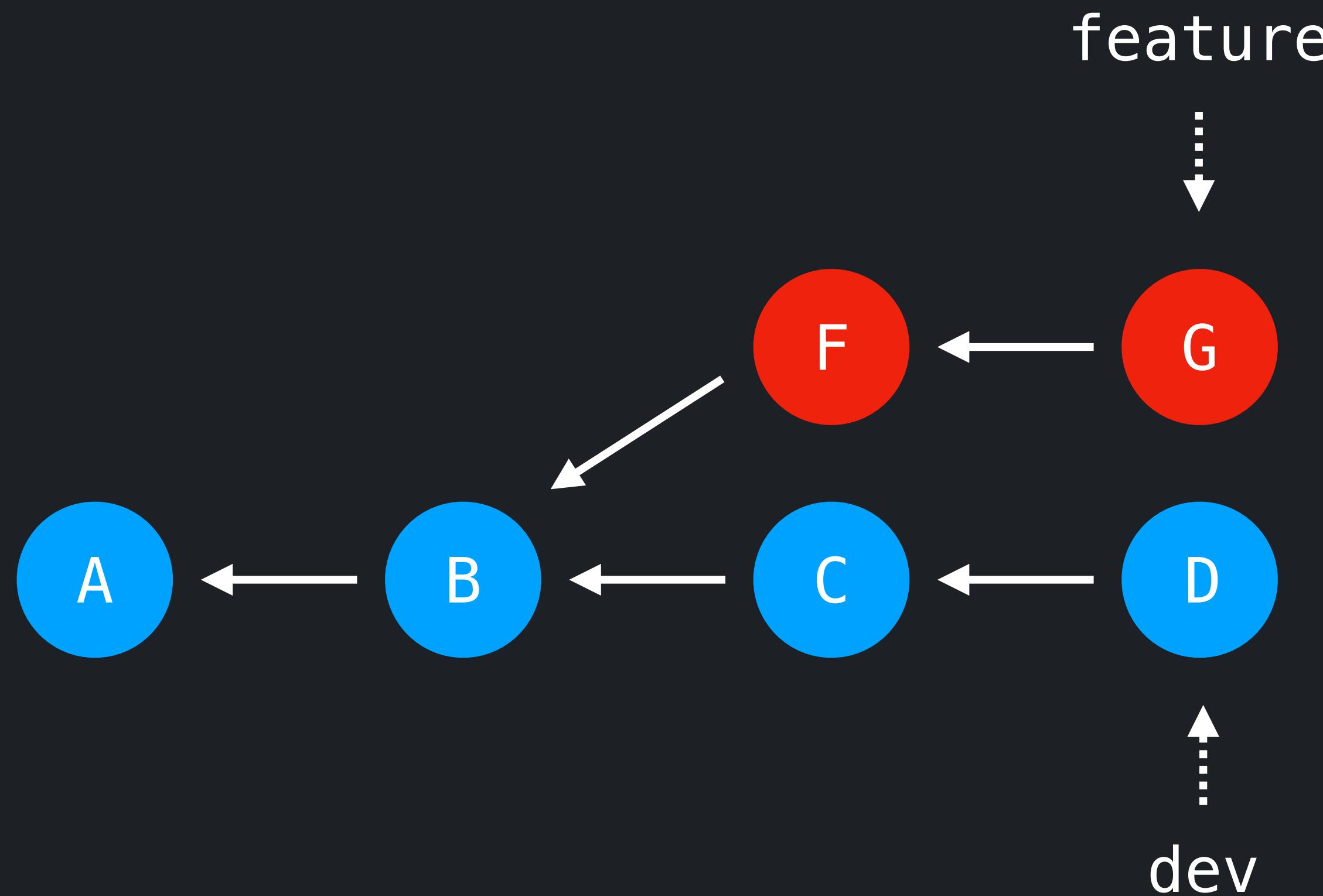
Merge



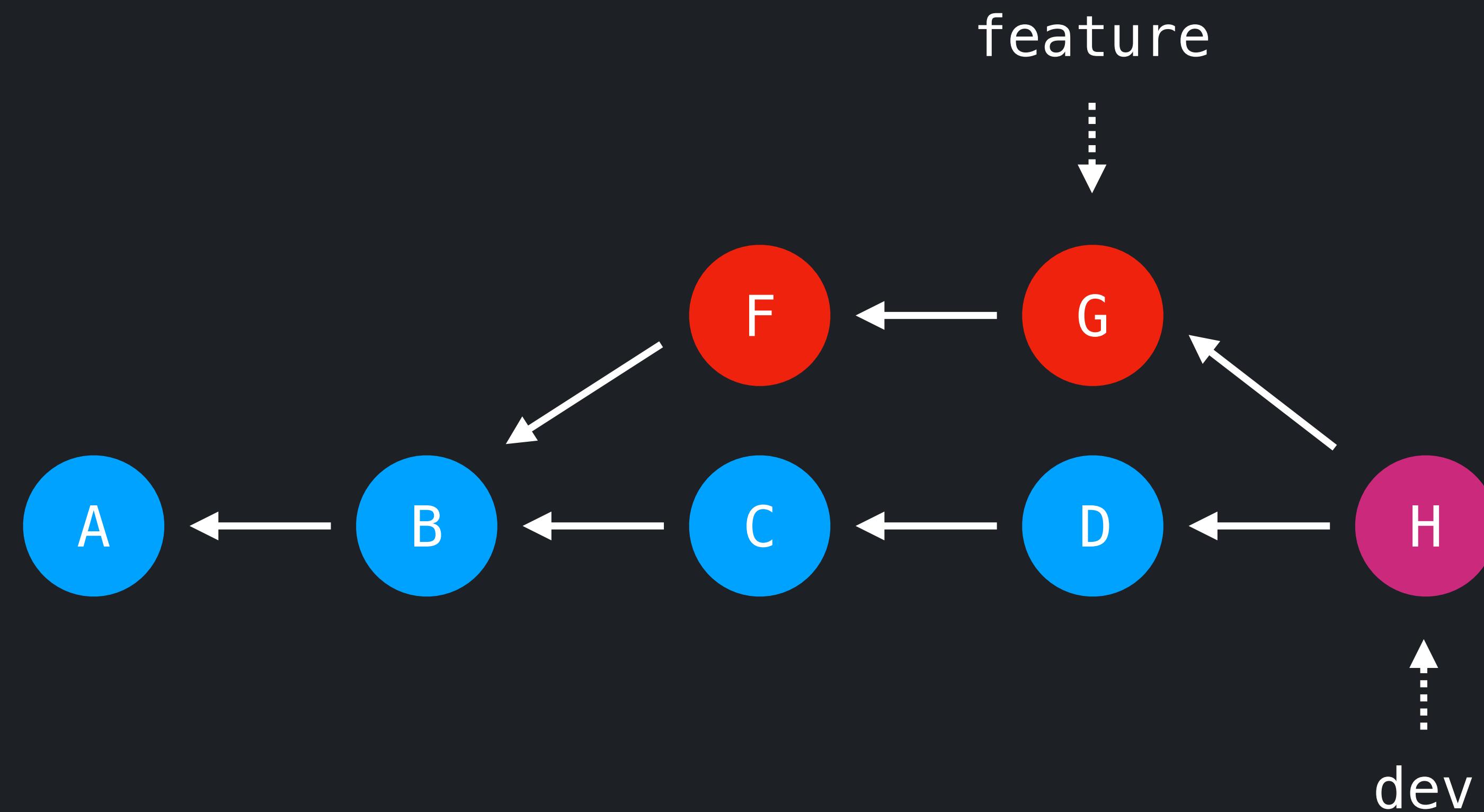
Merge



Merge

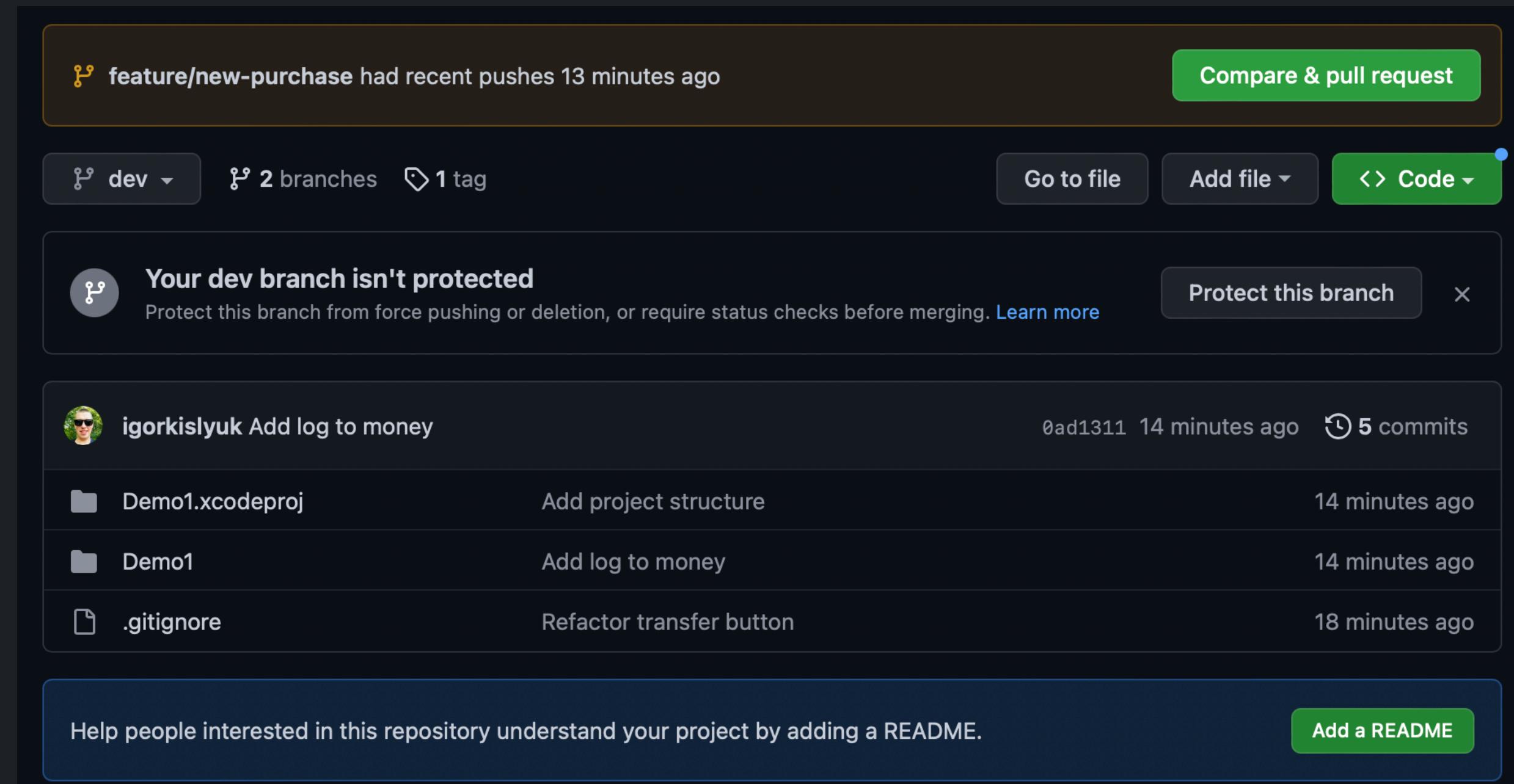


Merge



Merge Request

- отправляете ваши изменения на ревью
- Ситуация: хочу показать изменения коллегам



Merge conflict

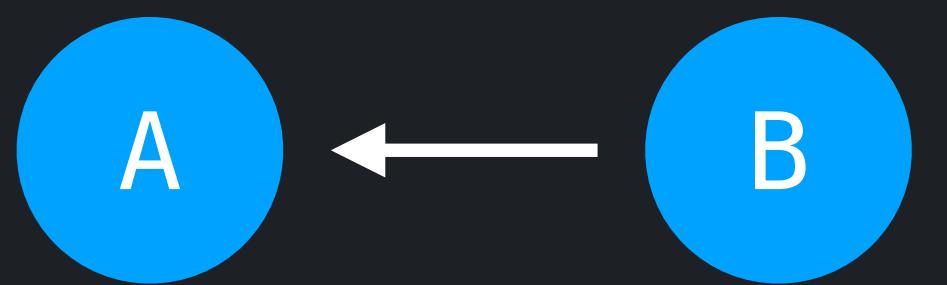
- выделяется в тексте <<<<<HEAD
- можно исправить руками
- `git add . && git commit`
- Ситуация: параллельные изменения

Rebase

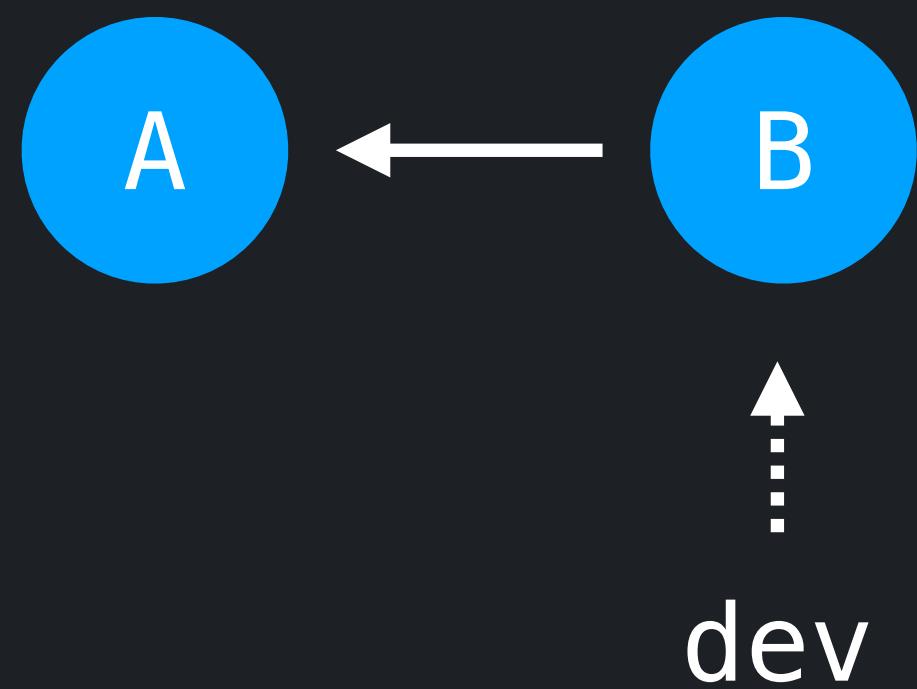
- добавляет изменения
- добавляет ваши коммиты сверху
- может быть интерактивным
- `git rebase BRANCH`
- Ситуация: скачиваю последние изменения в свою ветку

Rebase

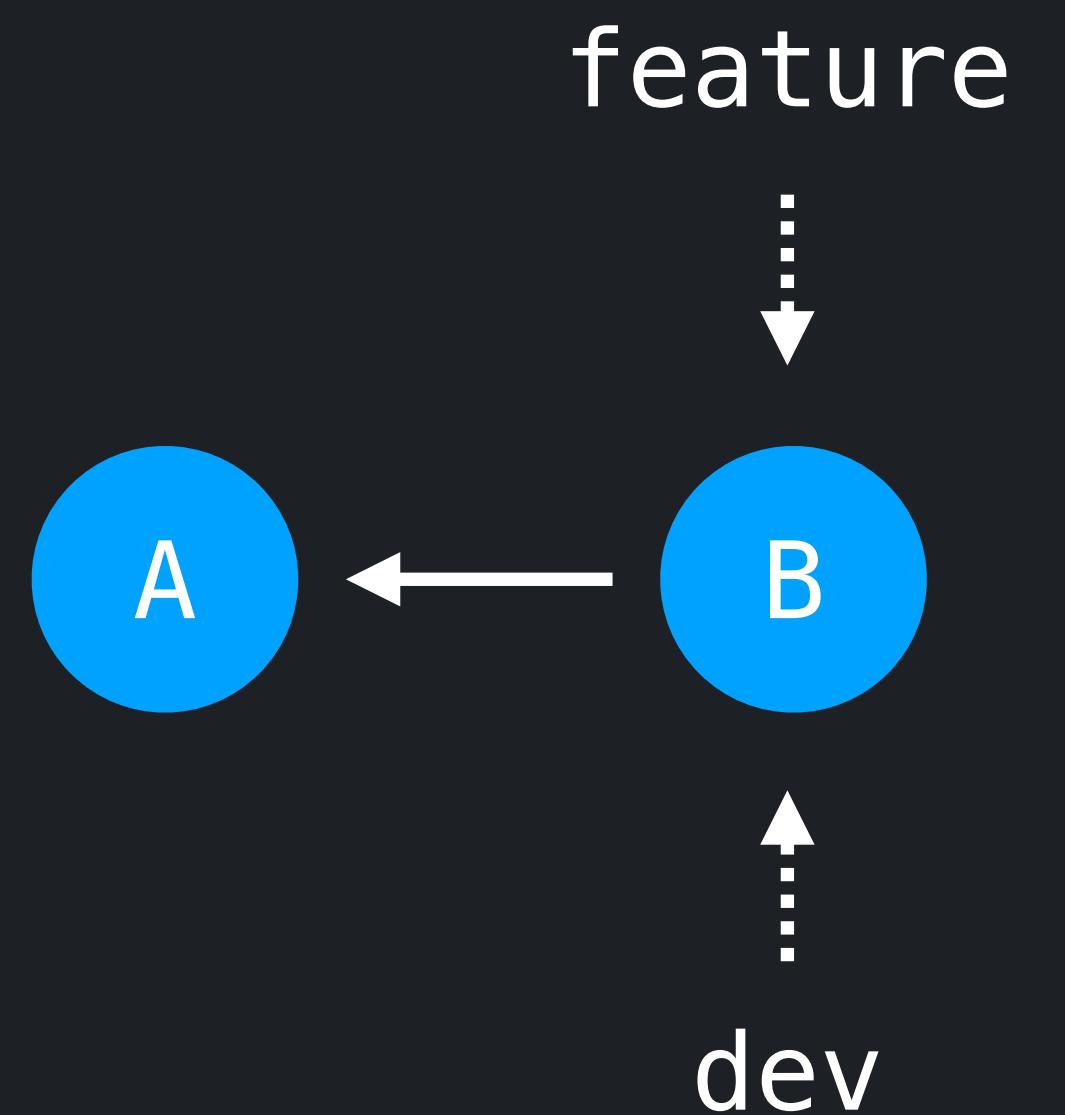
Rebase



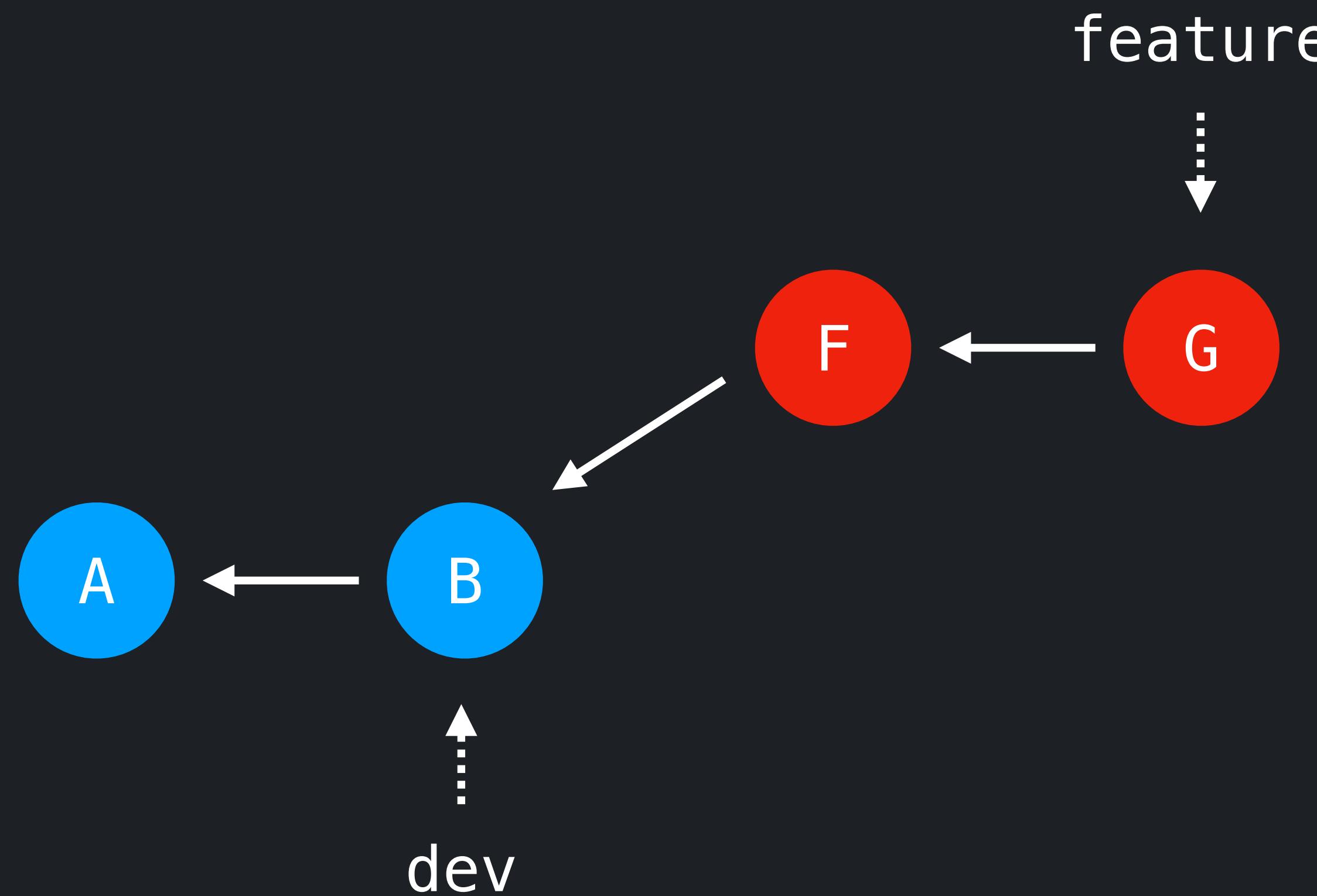
Rebase



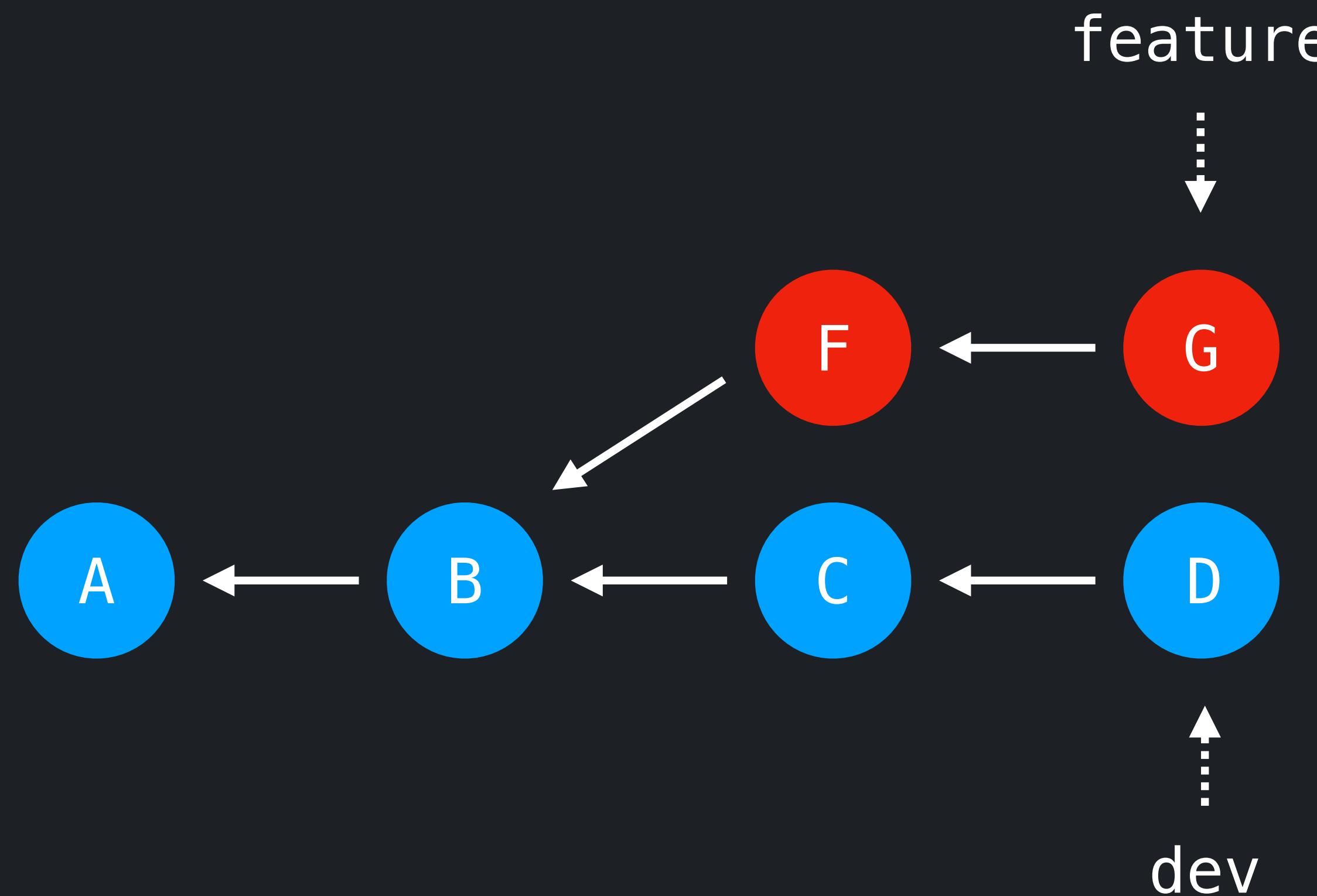
Rebase



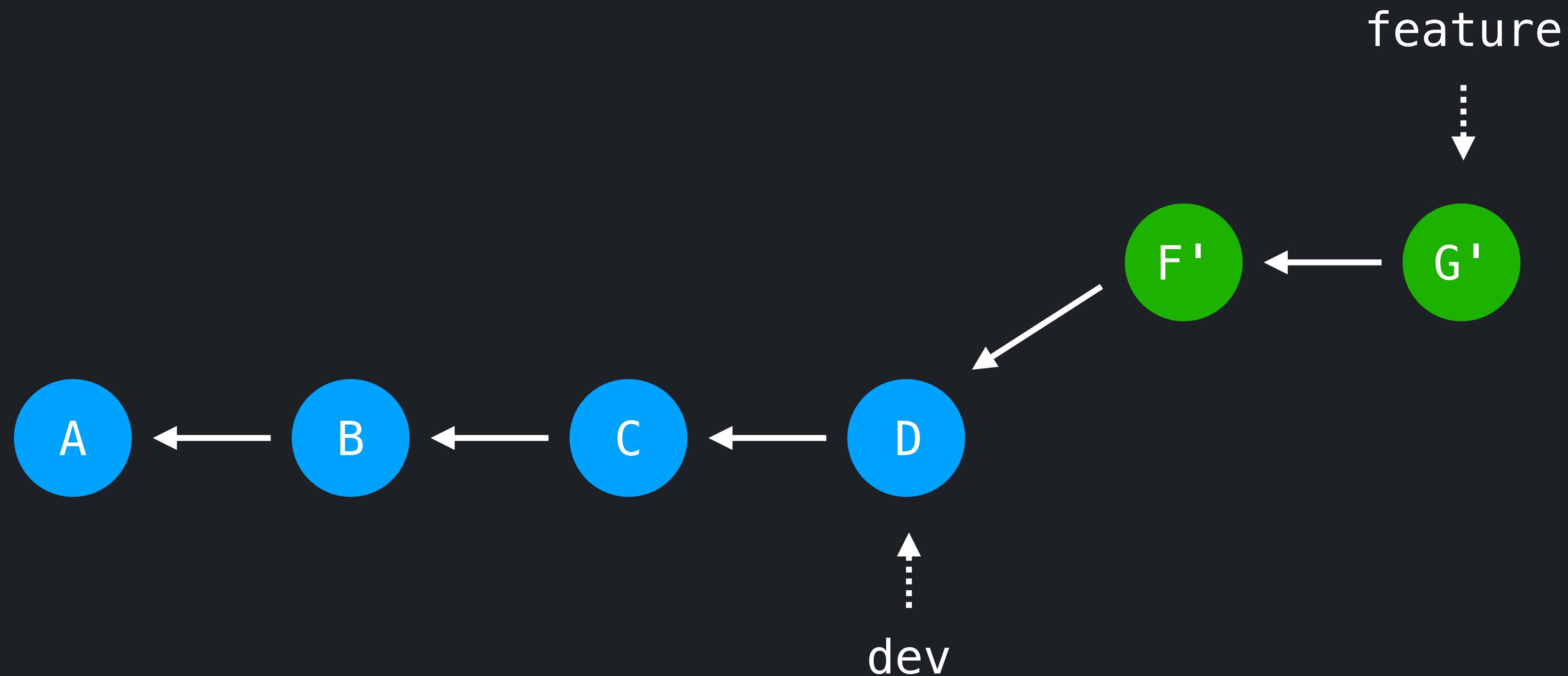
Rebase



Rebase



Rebase

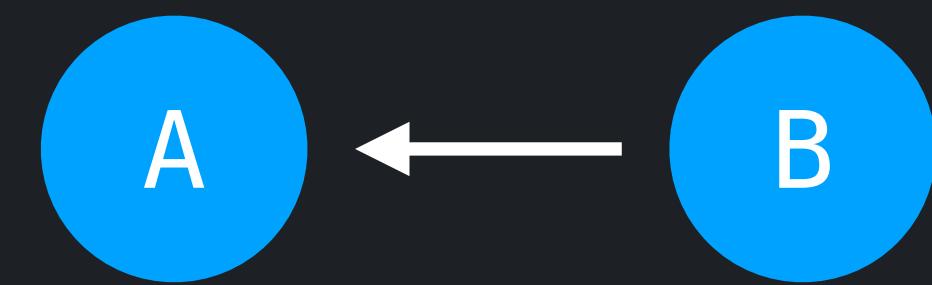


Pull

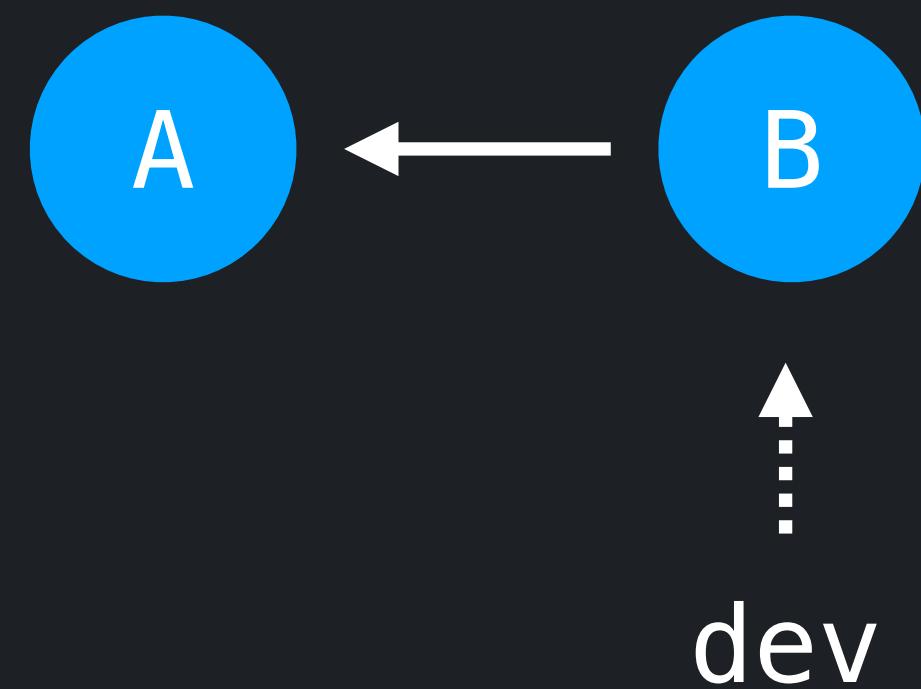
- `git pull = git fetch && git merge`
- загружает изменения
- меняет локальные указатели

Non Fast-Forward problem

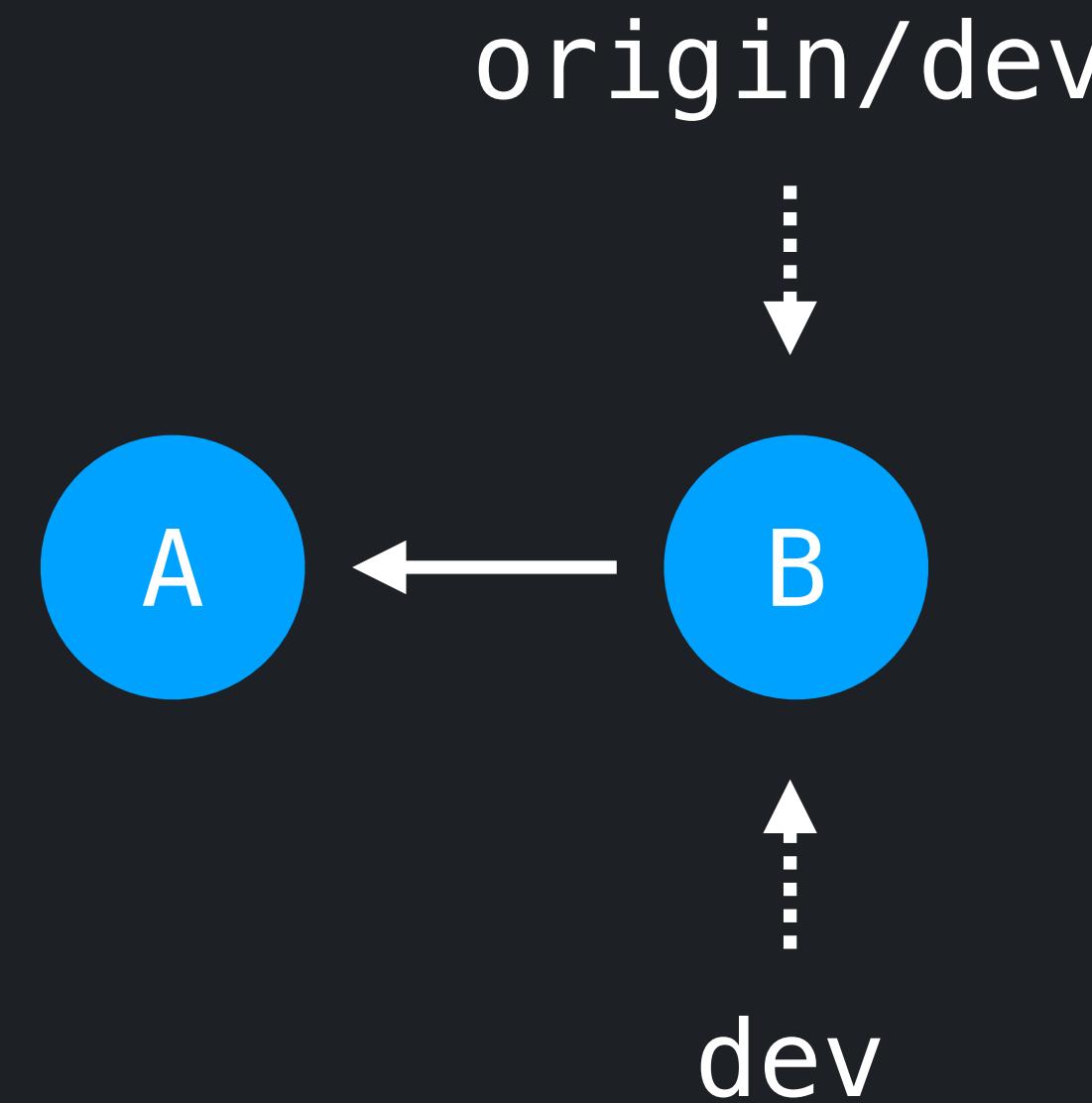
Non Fast-Forward problem



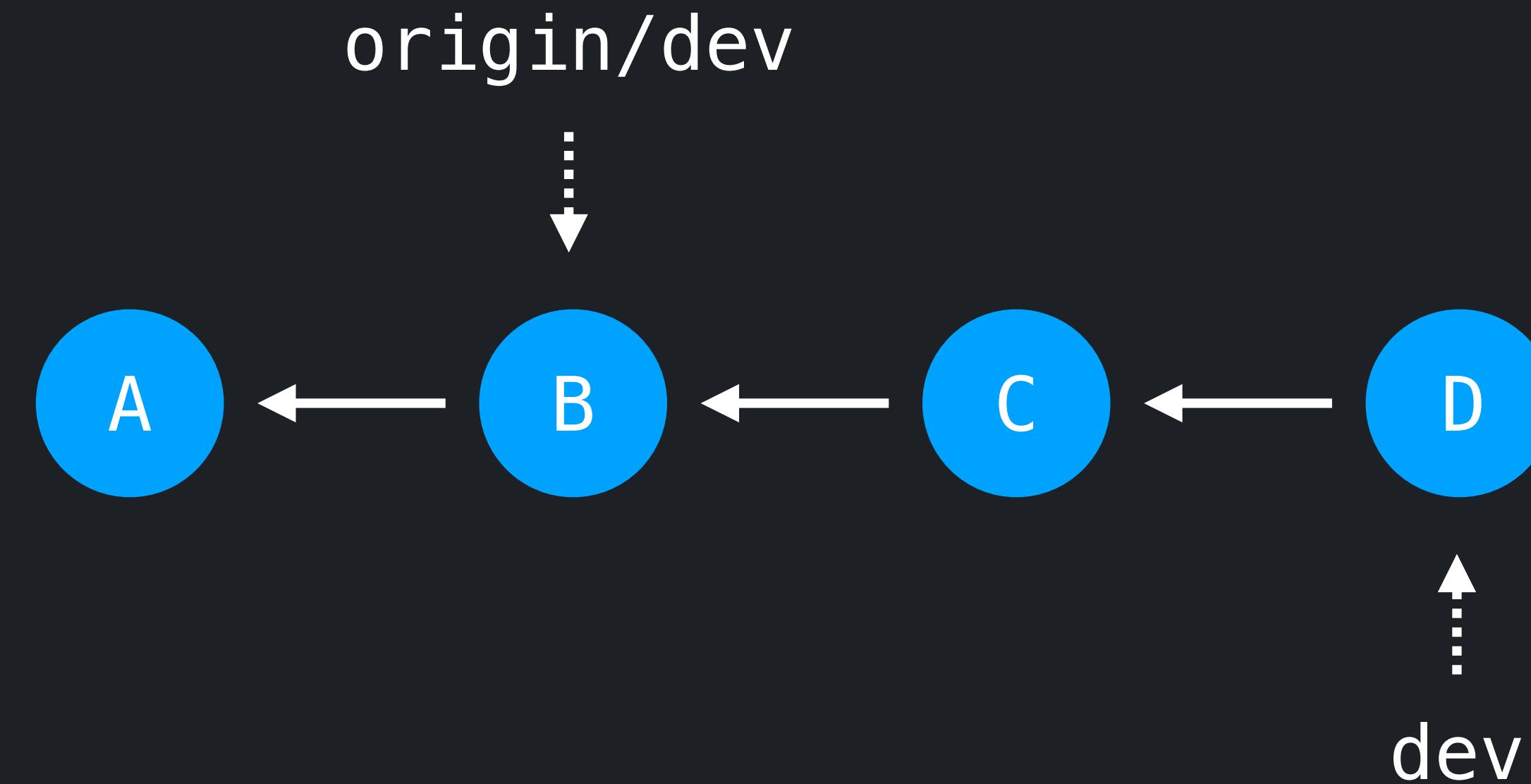
Non Fast-Forward problem



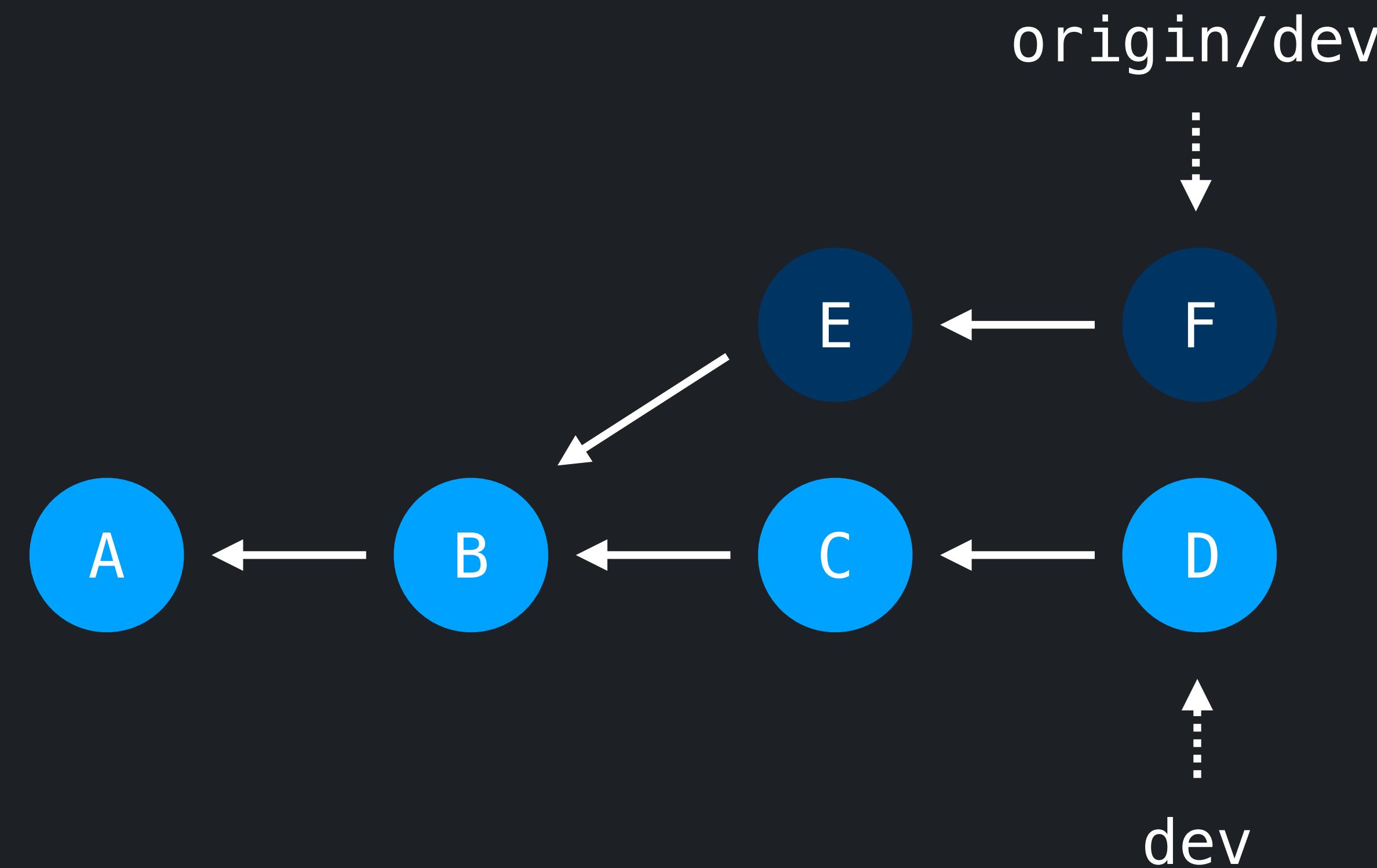
Non Fast-Forward problem



Non Fast-Forward problem



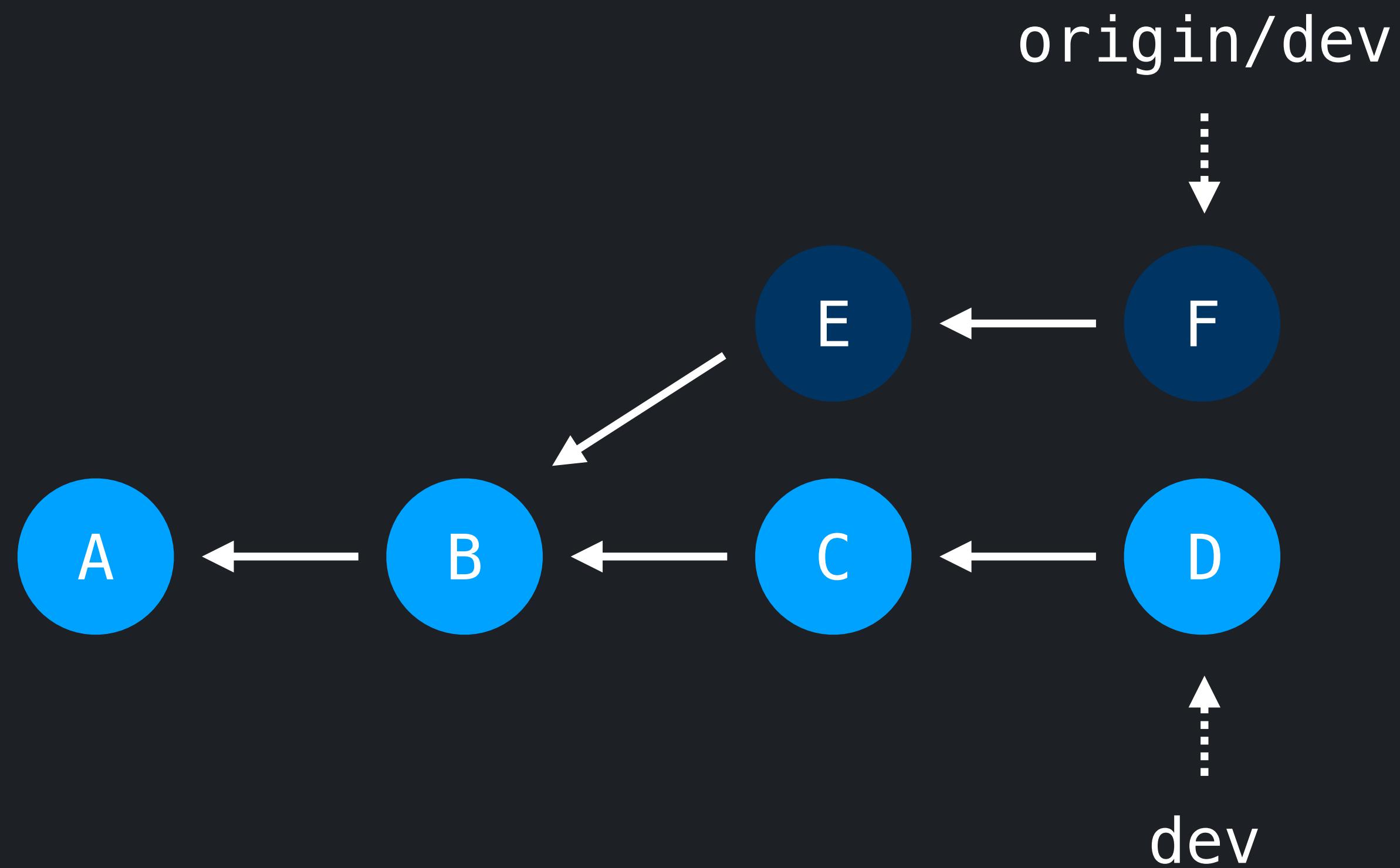
Non Fast-Forward problem



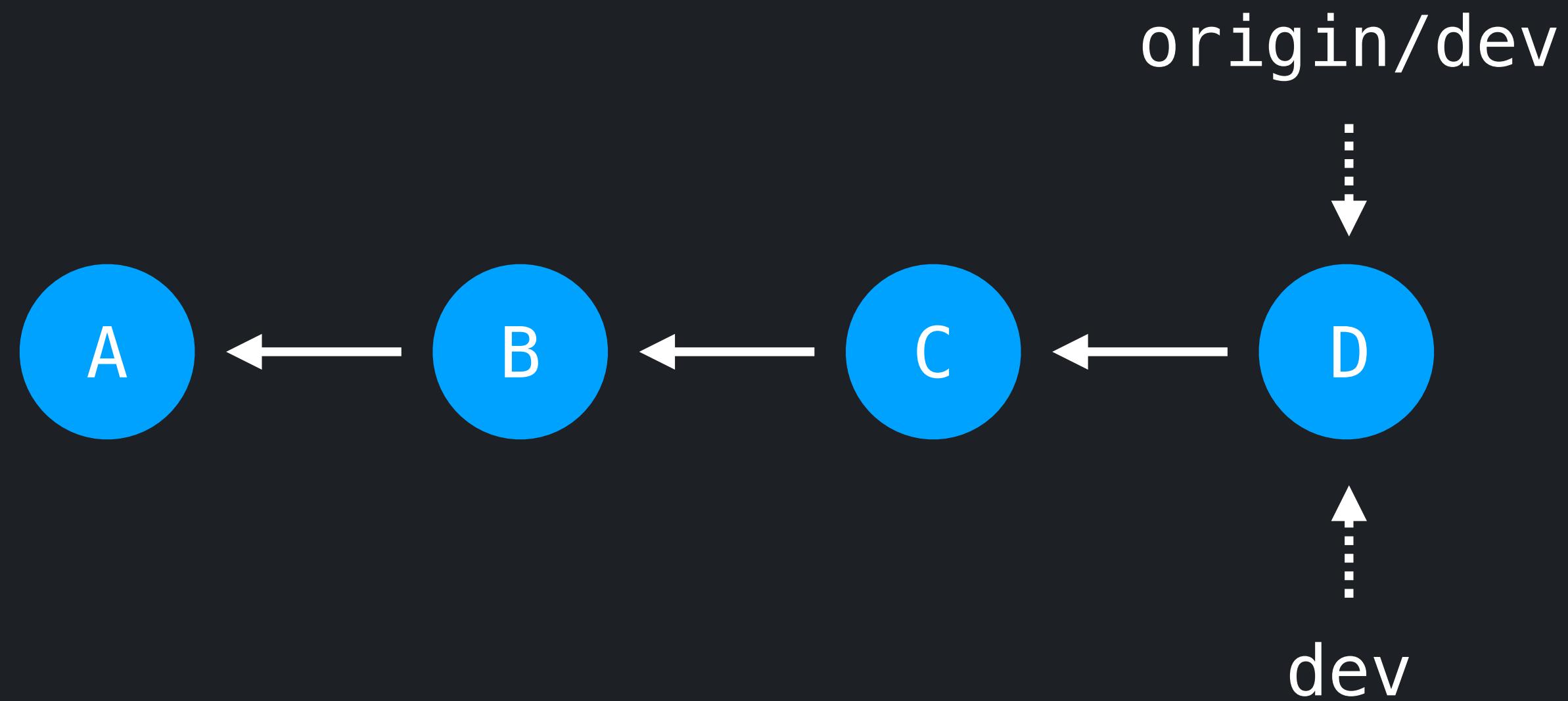
Решение Non Fast-Forward

- "откидывание изменений" (push --force)
- "интеграция изменений" (rebase, merge)

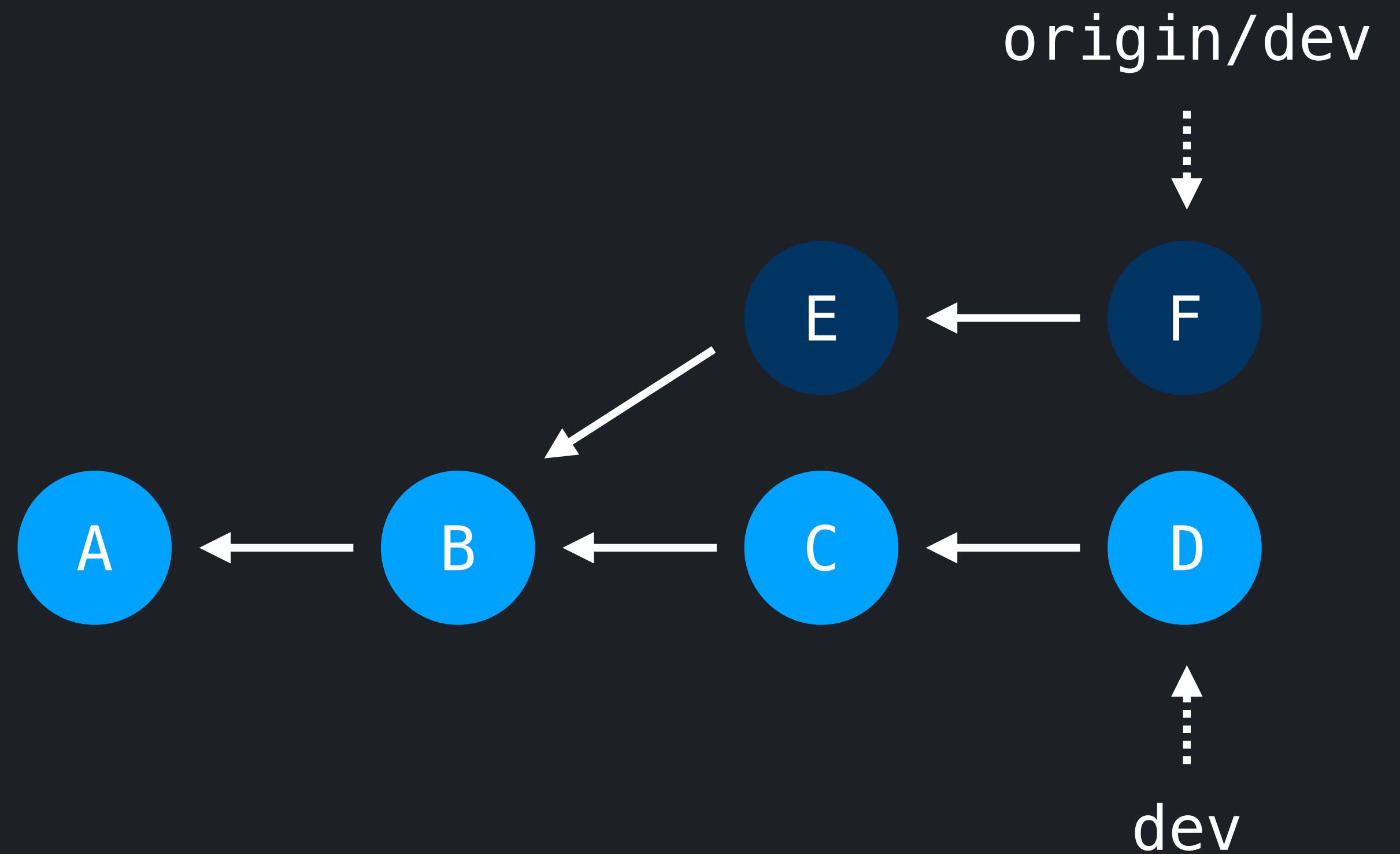
Force Push



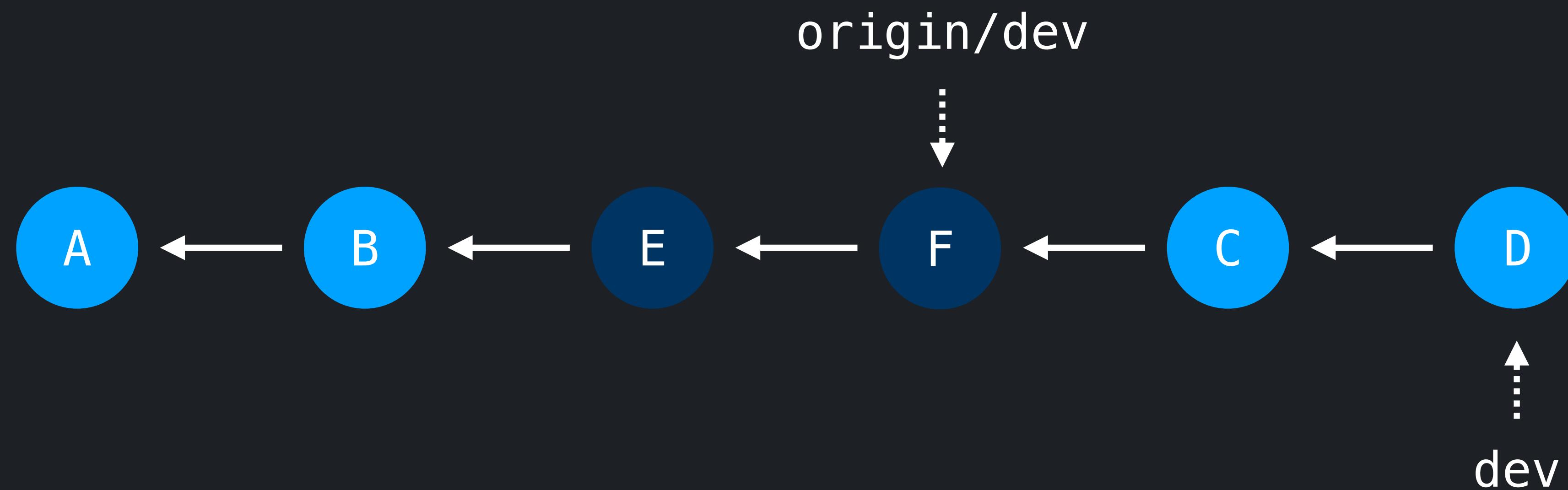
Force Push



Pull



Pull



GIT – GUI

Групповая динамика и коммуникации в профессиональной практике
программной инженерии

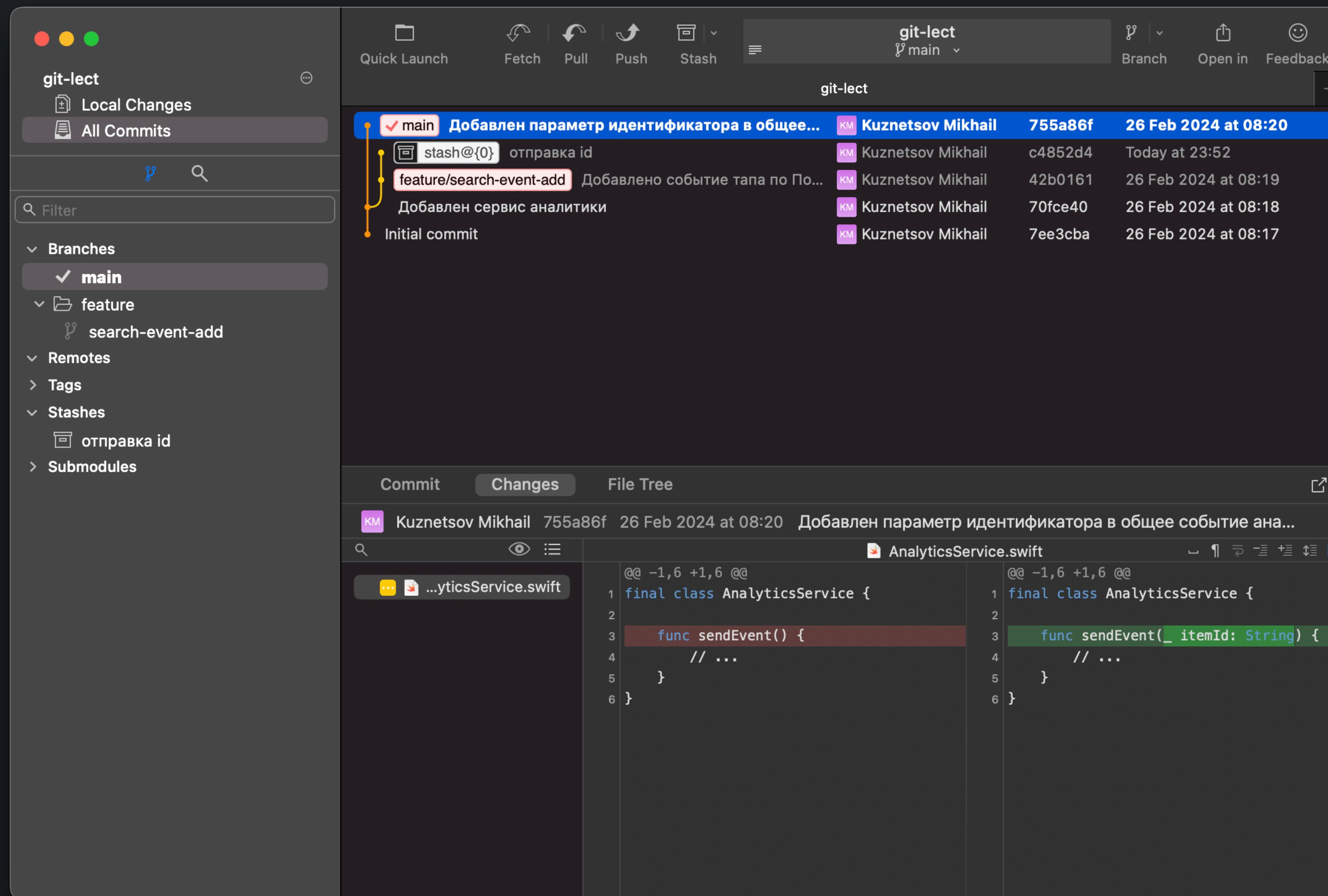
AnalyticsService.swift

...

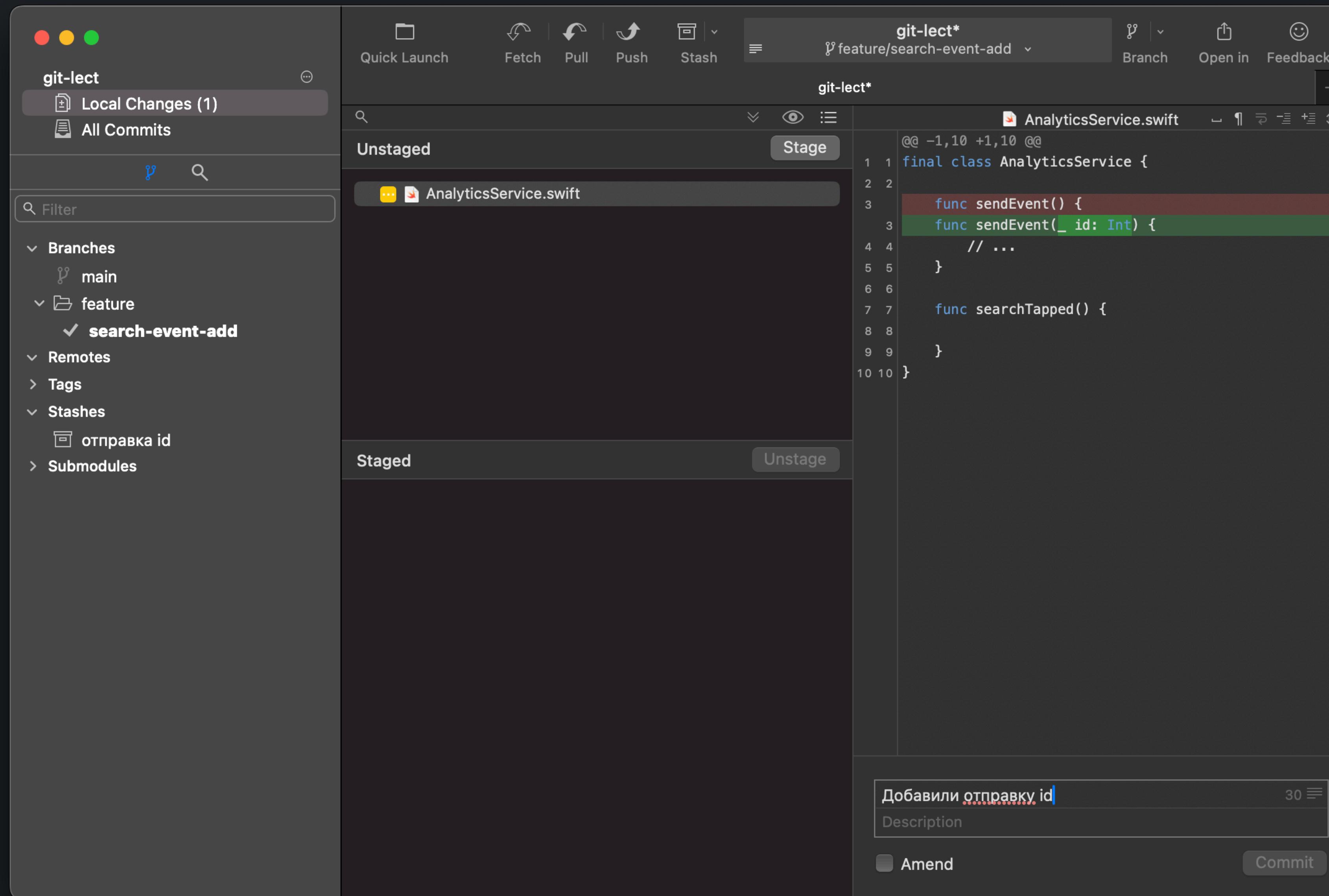
```
final class AnalyticsService {

    func sendEvent() {
        // ...
    }
}
```

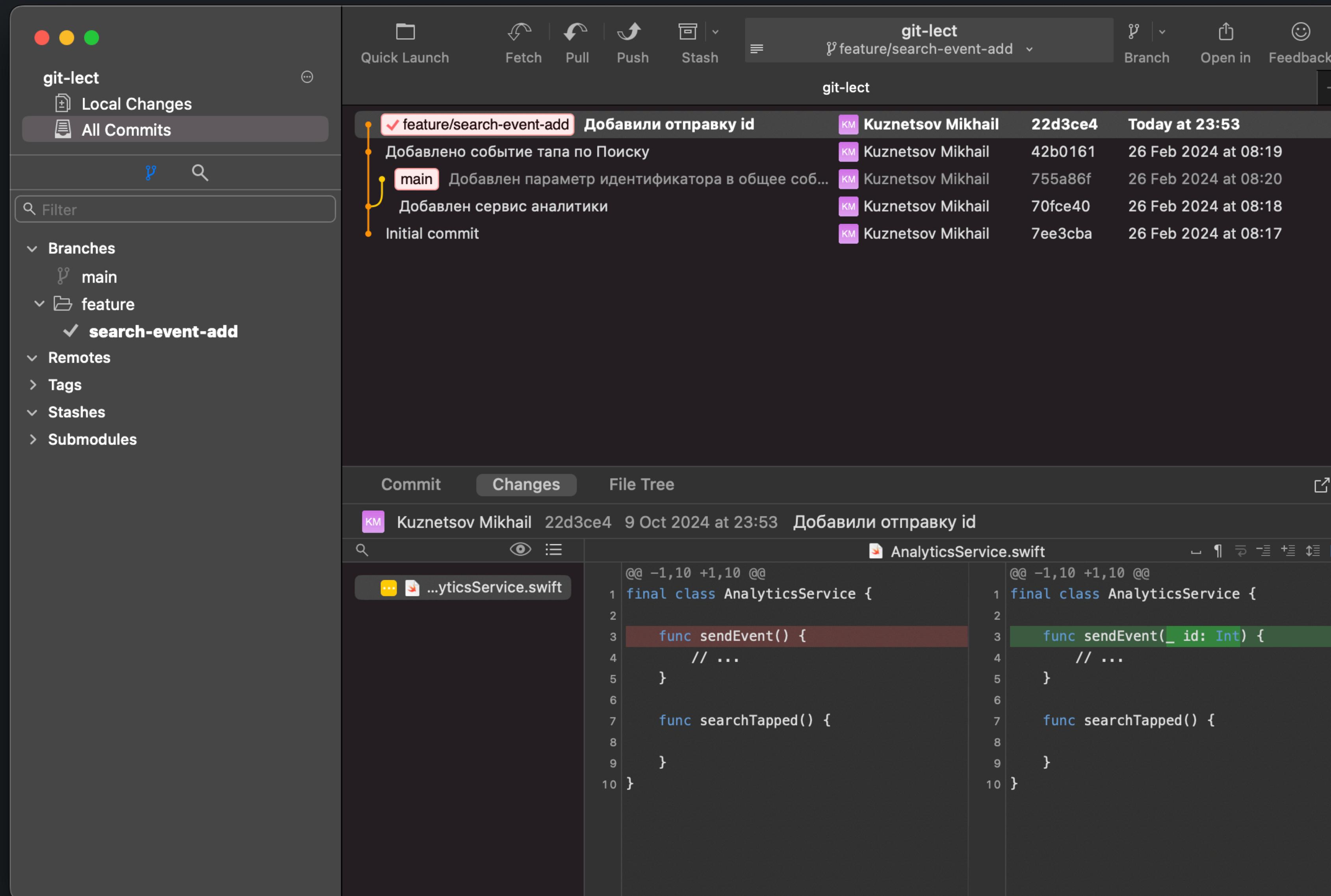
Fork



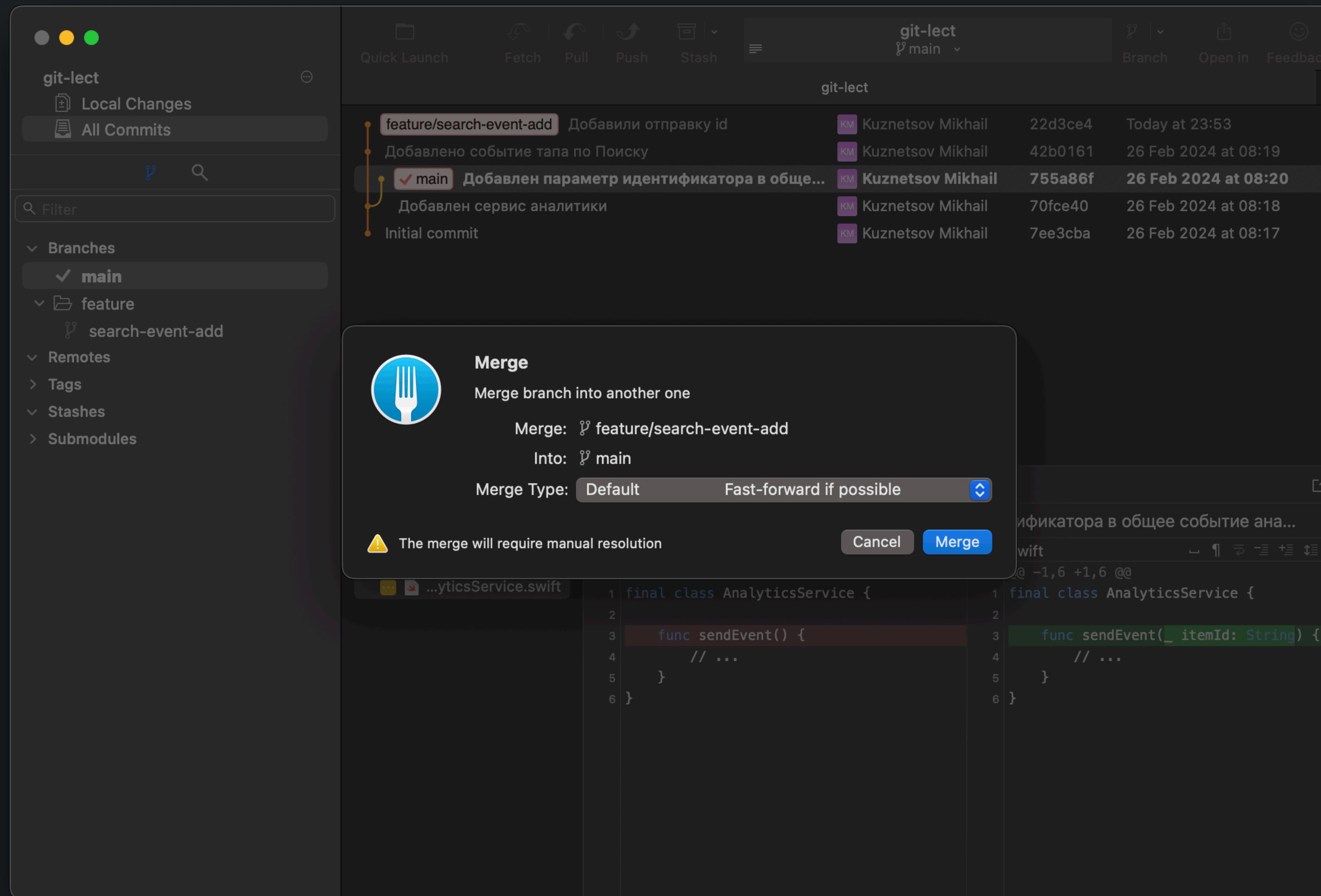
Fork I вносим изменения



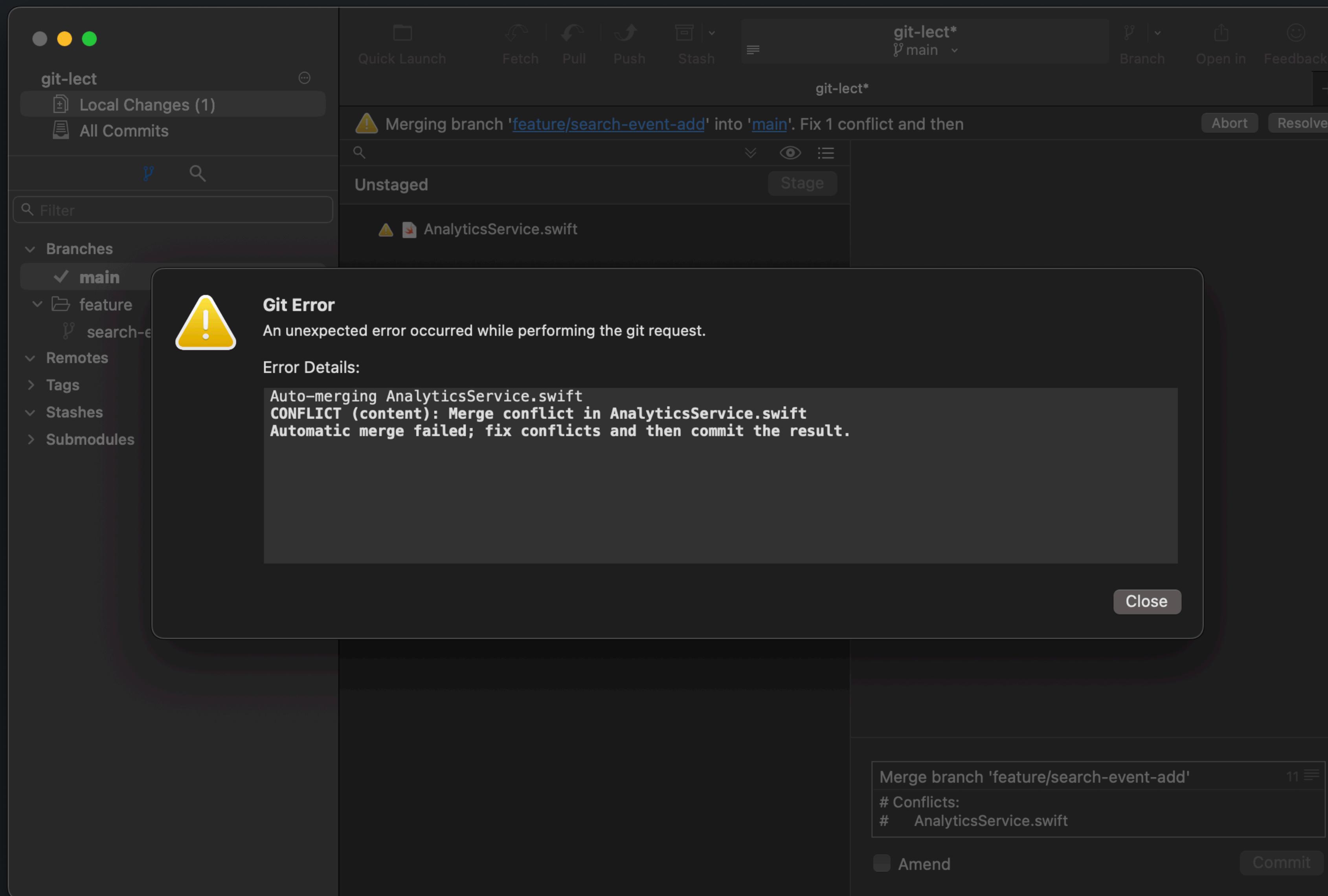
Fork



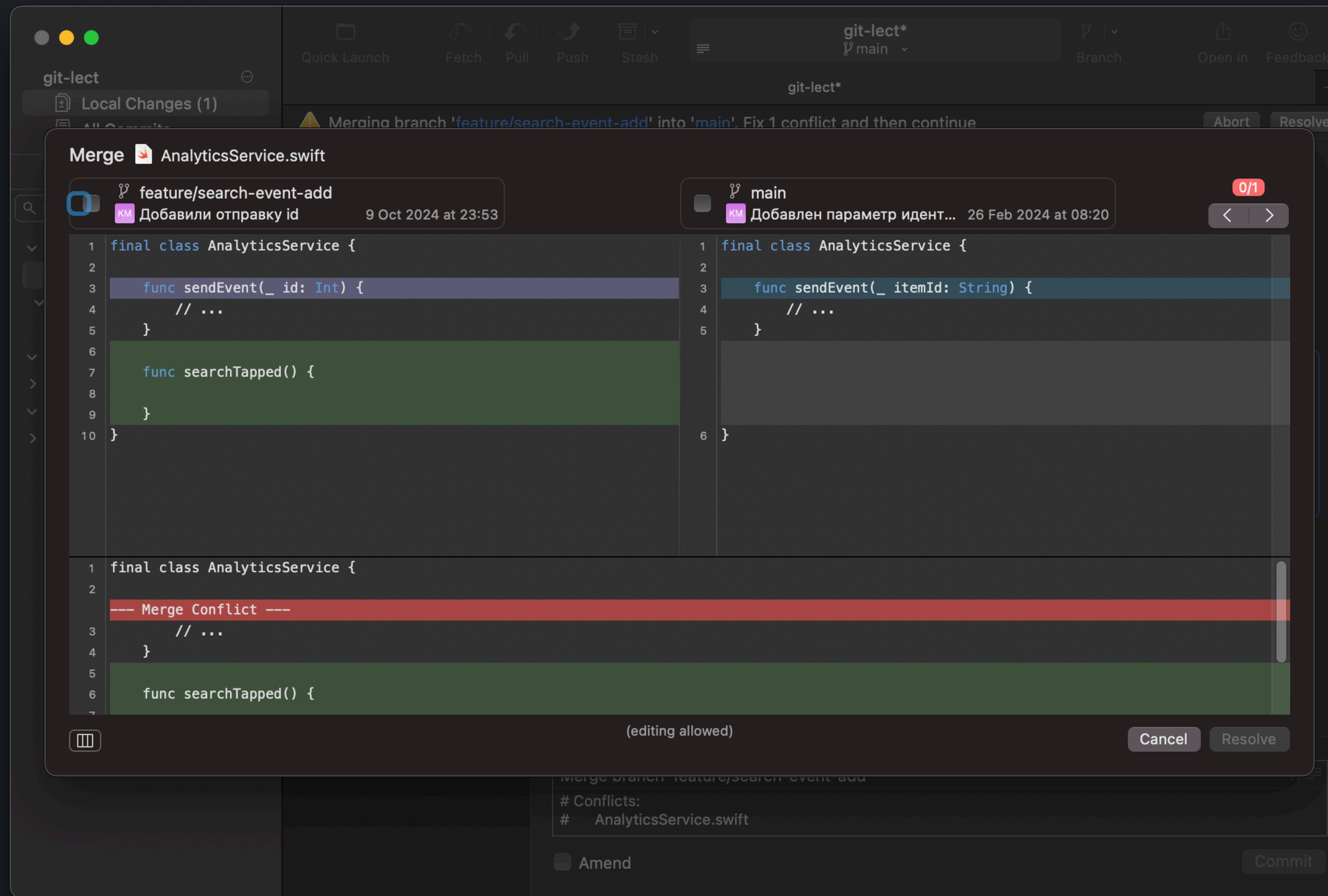
Fork I влияем нашу фичу в майн



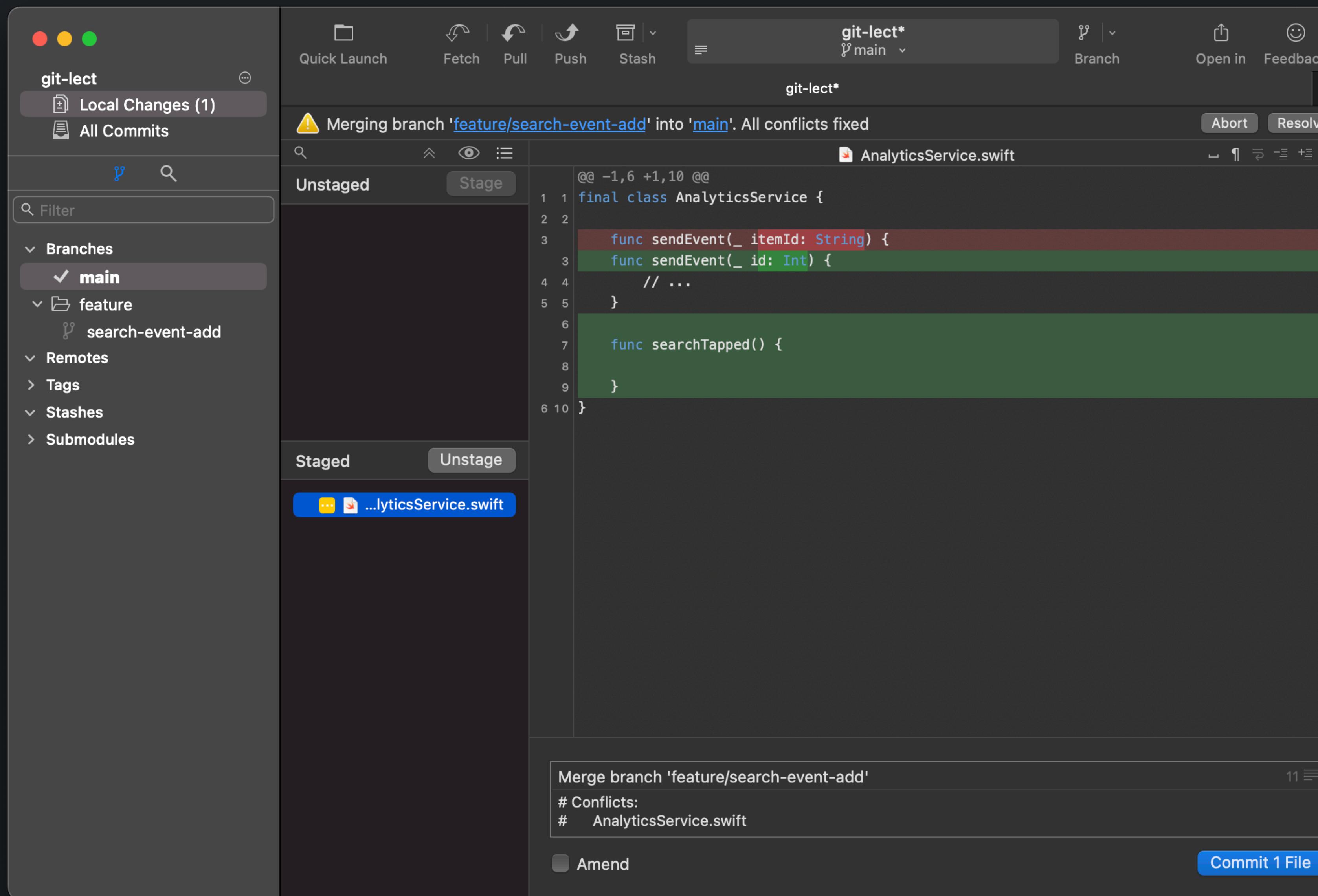
Fork I ловим конфликт



Fork I вручную разрешим конфликт



Fork I коммитим мердж



Fork I любуемся красотой

The screenshot shows a dark-themed Git interface with the following details:

Branch: main

Commit: Merge branch 'feature/search-event-add' (by Kuznetsov Mikhail, 8ef53ac, Today at 23:57)

Changes:

File	Line	Change Type	Text
AnalyticsService.swift	1	@@ -1,6 +1,10 @@	final class AnalyticsService {
AnalyticsService.swift	3	+ func sendEvent(_ itemId: String) {	func sendEvent(_ id: Int) {
AnalyticsService.swift	4	// ...	// ...
AnalyticsService.swift	6	+ func searchTapped() {	
AnalyticsService.swift	10	}	}

Commits:

- ✓ main Merge branch 'feature/search-event-add' (Kuznetsov Mikhail, 8ef53ac, Today at 23:57)
 - feature/search-event-add Добавили отправку id
 - Добавлен событие тапа по Поиску
 - Добавлен параметр идентификатора в общее событие аналитики
 - Добавлен сервис аналитики
 - Initial commit
- Kuznetsov Mikhail 22d3ce4 Today at 23:53
- Kuznetsov Mikhail 42b0161 26 Feb 2024 at 08:19
- Kuznetsov Mikhail 755a86f 26 Feb 2024 at 08:20
- Kuznetsov Mikhail 70fce40 26 Feb 2024 at 08:18
- Kuznetsov Mikhail 7ee3cba 26 Feb 2024 at 08:17

Extra

Групповая динамика и коммуникации в профессиональной практике
программной инженерии

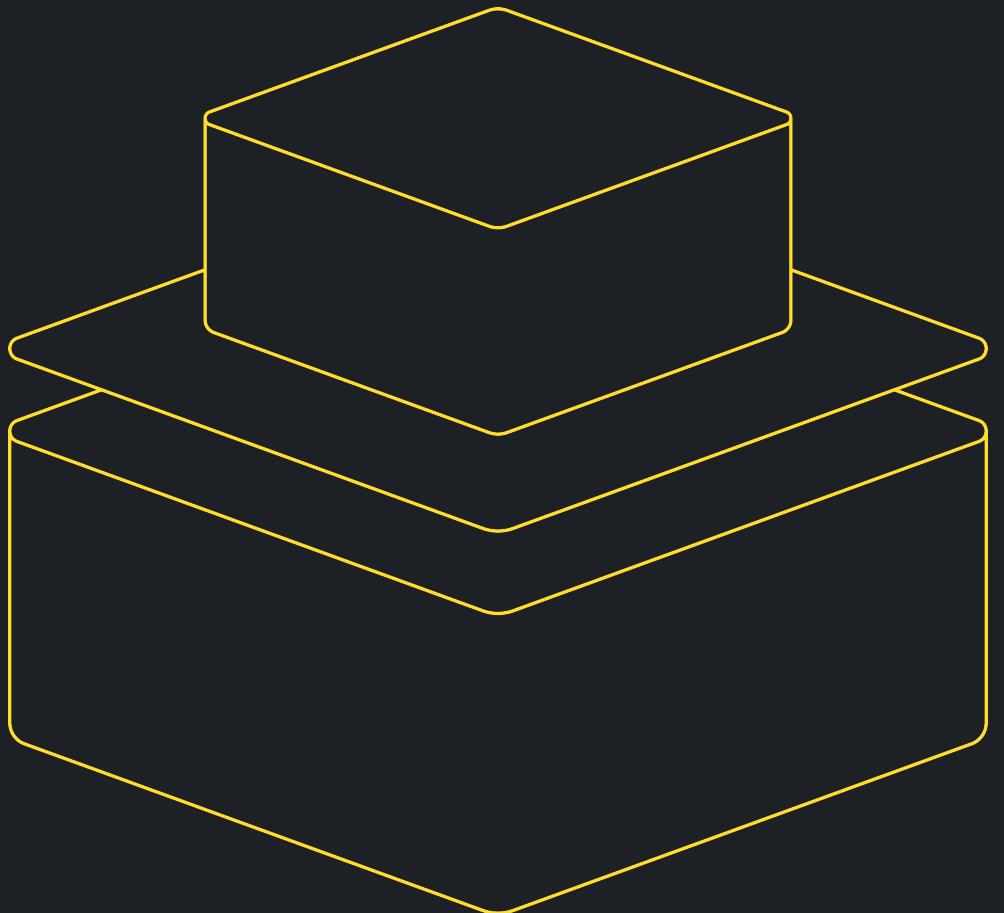
Случайно удалил ветку?

- корзина (на 30 дней)
- `git reflog`
- находите нужный коммит
- `git branch BRANCH SHA`



Git Hooks

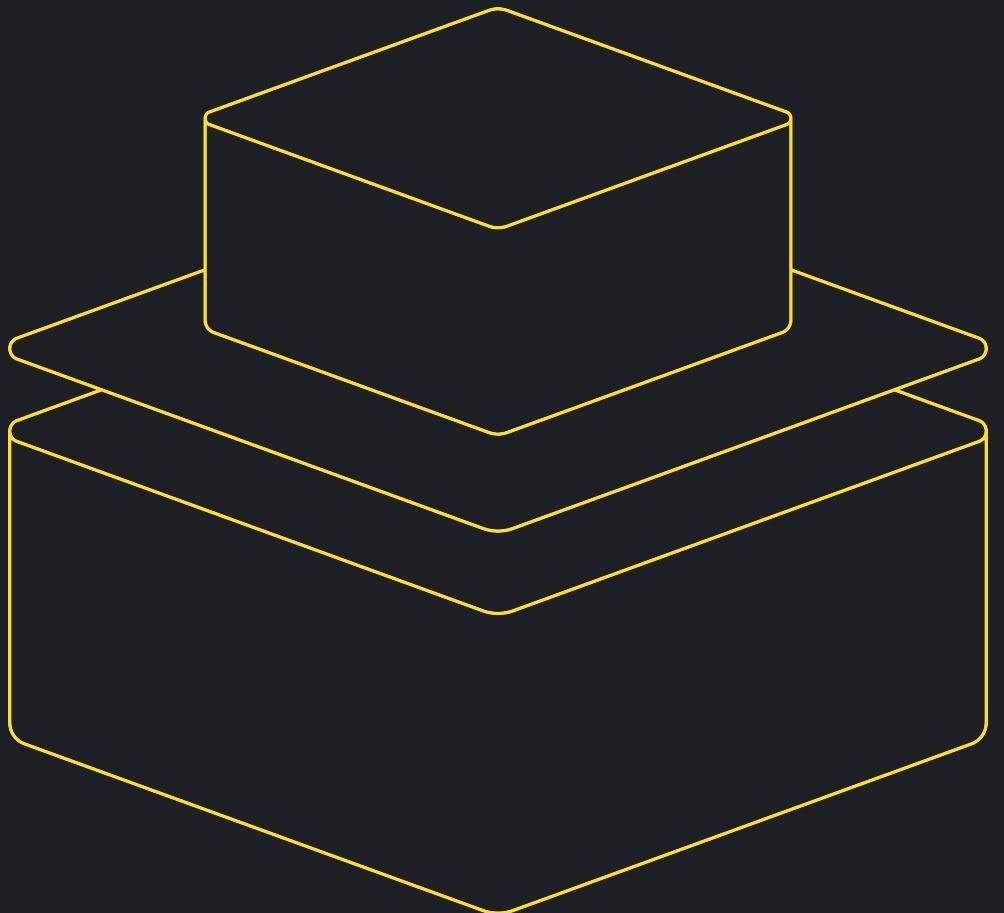
- вызываются после определенной команды
- нужны для скриптов
- `swiftlint` перед коммитом в `pre-commit`



Cherry Pick

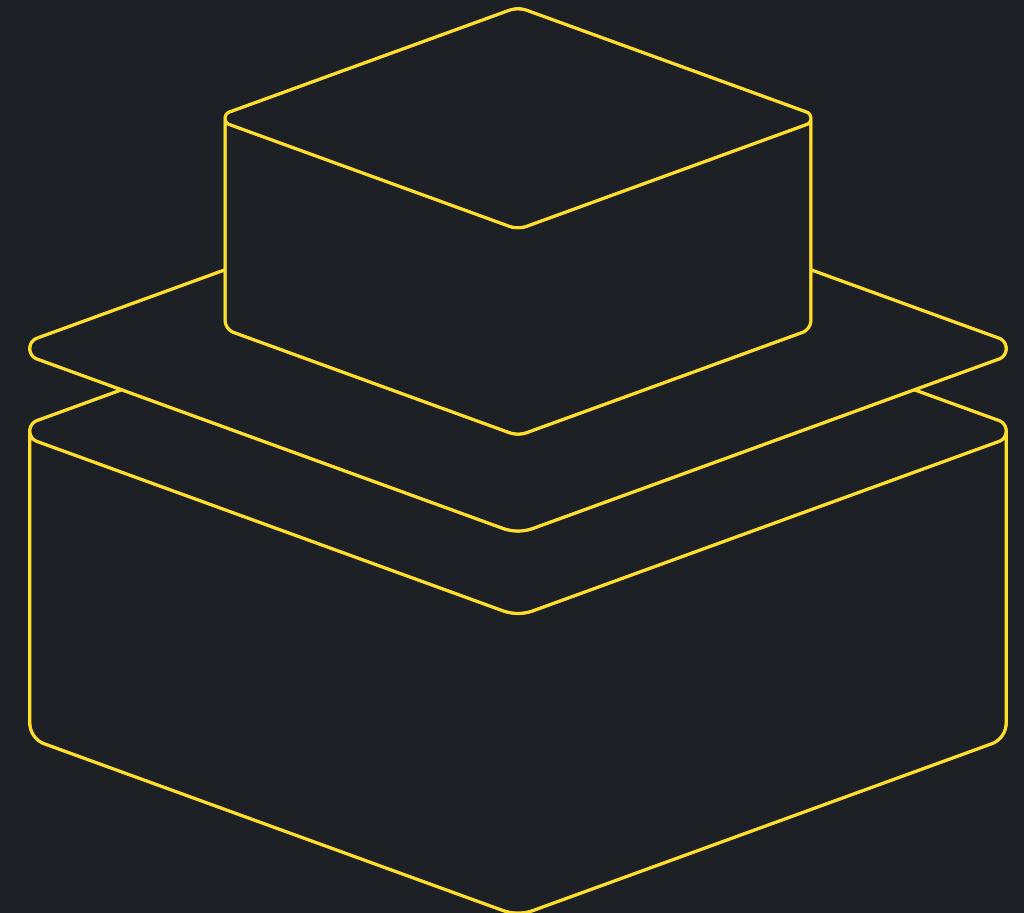


- переносим изменения между ветками
- используем для синхронизации dev и release



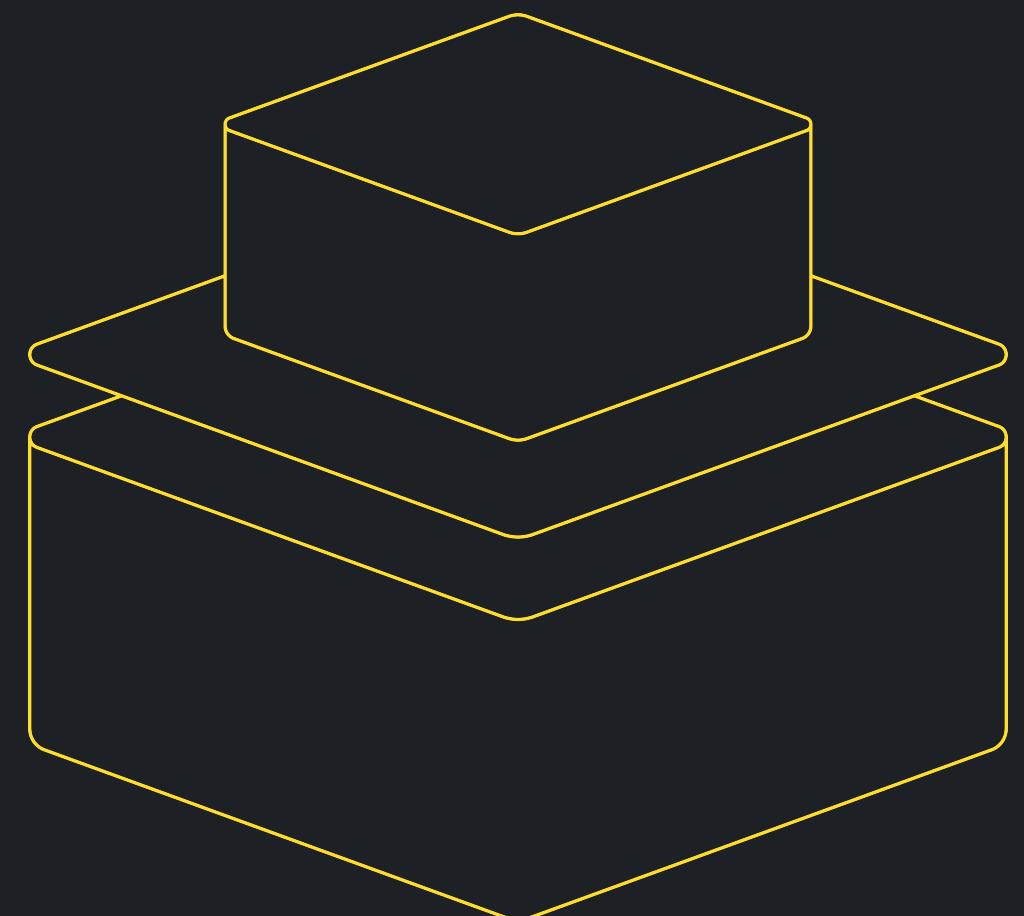
Ищем поломанный коммит

- бинарный поиск
- `git bisect start`
- `git bisect bad`
- `git bisect good SHA`



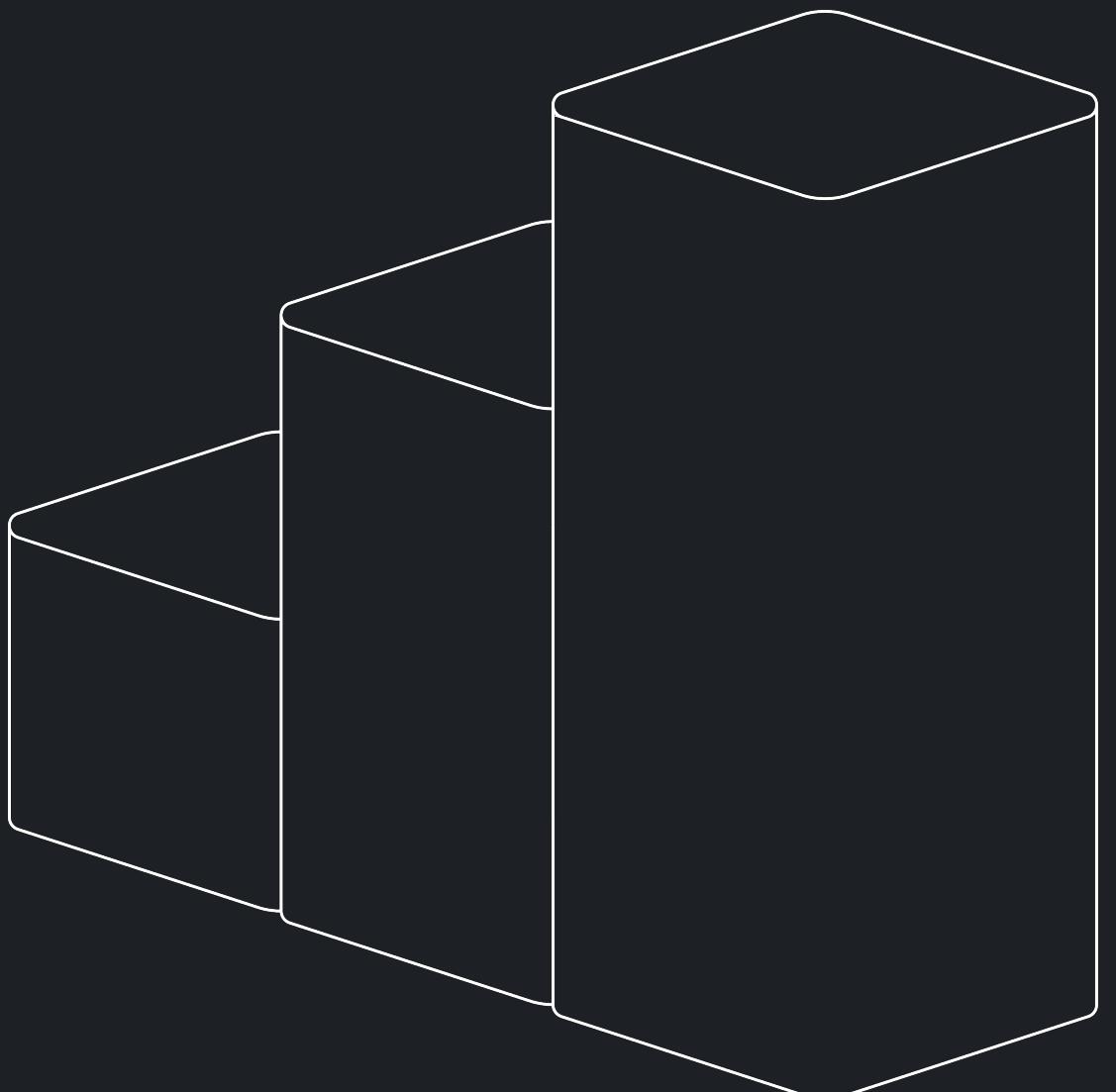
Храним большие бинарные файлы

- `git-lfs`
- Создает ссылки на большие файлы
- Хранит сами файлы отдельно



А что с безопасностью?

- не храните **пароли** под git
- есть инструменты, которые позволяют злоумышленнику найти в коммитах секретную информацию
- используйте ssh



Ссылки на ресурсы



[Fork, SourceTree, VSCode + GitLens, etc](#)



[Красивый терминал](#)



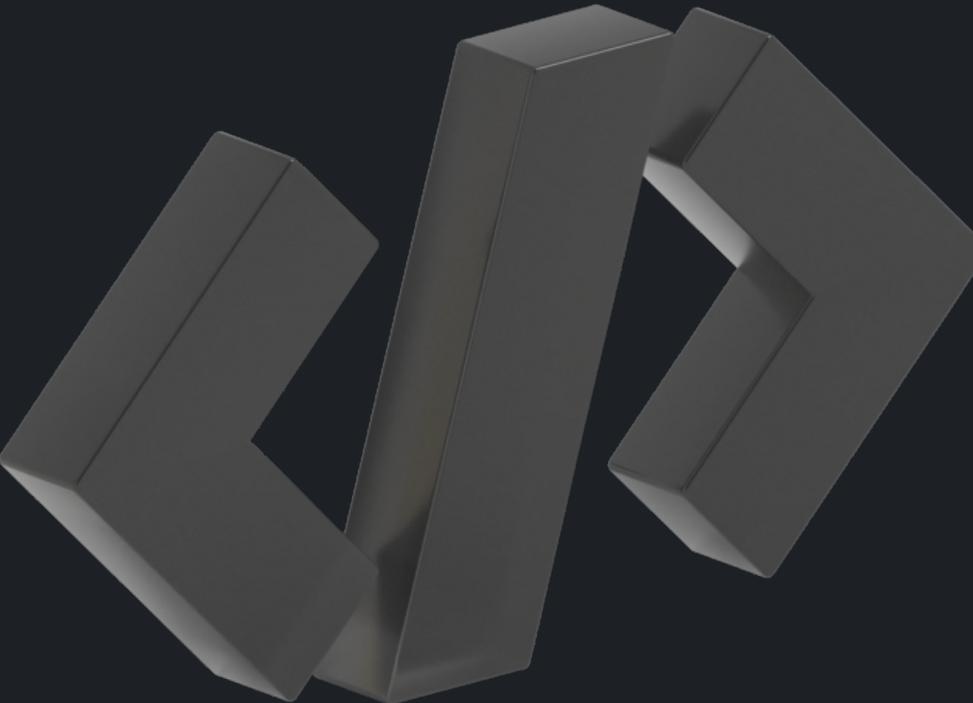
[Список gitignore](#)



[Документация](#)



[Инструкция по настройке SSH](#)



Спасибо за внимание !

Кузнецов Михаил Александрович
ma.kuznetsov@hse.ru
@rokymiel



факультет компьютерных наук