

DevOps

Л03. Системы контроля версий на примере Git
П03. git

Виктор Моисеев
+7-902-83-145-30
t.me/v_paranoid
victorparanoid@gmail.com

План курса

1. Введение в DevOps
2. Базовое администрирование Linux
3. Системы контроля версионности кода (git)
4. Оркестровка (Ansible)
5. Контейнеризация (docker)
6. Микросервисная архитектура и оркестровка контейнеров (k8s)
7. Непрерывная интеграция и доставка (CI/CD, Github Actions, ArgoCD)
8. Инфраструктура как код (IaC, Terraform)
9. Мониторинг (Prometheus)

03. git

1. Что такое git
2. Принцип работы (diff, 3-way)
3. Перед началом работы (установка и настройка)
4. Работа с локальным репозиторием, инициализация
5. Работа с удаленным репозиторием, протоколы
6. Ветки
7. Способы слияния (ff/merge/rebase)
8. Защита веток
9. Внешние инструменты (merge/pull requests)
10. Теги
11. Cherry-pick

<https://git-scm.com/book/en/v2/>

<https://www.atlassian.com/ru/git>

Что такое (D)VCS

Изменения в коде

Версии, ревизии

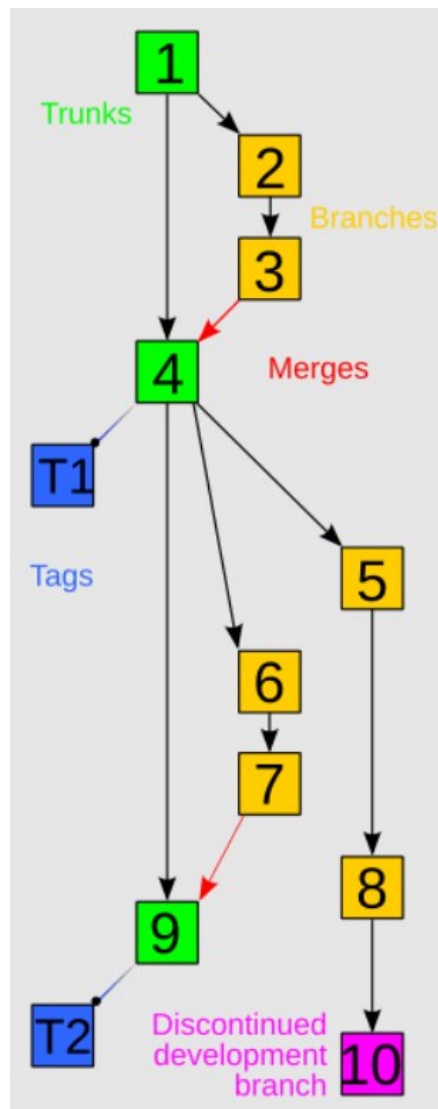
Ветки (форк), слияния

Хранятся на сервере или распределенно

git

Mercurial

Subversion



Diff

Diff – ищет наиболее длинные общие последовательности текста – отмечает их позиции по номерам строк

original:

```
1 This part of the
2 document has stayed the
3 same from version to
4 version. It shouldn't
5 be shown if it doesn't
6 change. Otherwise, that
7 would not be helping to
8 compress the size of the
9 changes.
10
11 This paragraph contains
12 text that is outdated.
13 It will be deleted in the
14 near future.
15
16 It is important to spell
17 check this dokument. On
18 the other hand, a
19 misspelled word isn't
20 the end of the world.
21 Nothing in the rest of
22 this paragraph needs to
23 be changed. Things can
24 be added after it.
```

new:

```
1 This is an important
2 notice! It should
3 therefore be located at
4 the beginning of this
5 document!
6
7 This part of the
8 document has stayed the
9 same from version to
10 version. It shouldn't
11 be shown if it doesn't
12 change. Otherwise, that
13 would not be helping to
14 compress the size of the
15 changes.
16
17 It is important to spell
18 check this document. On
19 the other hand, a
20 misspelled word isn't
21 the end of the world.
22 Nothing in the rest of
23 this paragraph needs to
24 be changed. Things can
25 be added after it.
```

The command `diff original new` produces the following *normal diff output*:

```
0a1,6
> This is an important
> notice! It should
> therefore be located at
> the beginning of this
> document!
>
11,15d16
< This paragraph contains
< text that is outdated.
< It will be deleted in the
< near future.
<
17c18
< check this dokument. On
---
> check this document. On
24a26,29
>
> This paragraph contains
> important new additions
> to this document.
```

Diff контекстный

```
--- /path/to/original    timestamp
+++ /path/to/new        timestamp
@@ -1,3 +1,9 @@
+This is an important
+notice! It should
+therefore be located at
+the beginning of this
+document!
+
  This part of the
  document has stayed the
  same from version to
@@ -8,13 +14,8 @@
  compress the size of the
  changes.

-This paragraph contains
-text that is outdated.
-It will be deleted in the
-near future.
-
  It is important to spell
-check this dokument. On
+check this document. On
  the other hand, a
  misspelled word isn't
  the end of the world.
@@ -22,3 +23,7 @@
  this paragraph needs to
  be changed. Things can
  be added after it.
+
+This paragraph contains
+important new additions
+to this document.
```

Patch

Patch – применяет изменения, описанные в синтаксисе diff (ed script или unidiff)

unidiff format

```
$ diff -u oldFile newFile > mods.diff
```

```
$ patch < mods.diff
```

reversed

```
$ patch -R < mods.diff
```

Patch – может сориентироваться, если контекст переместили или изменили (fuzz)

git

<https://git-scm.com/book/>

<https://learngitbranching.js.org/>

2005 (Torvalds)

Speed

Simple design

Strong support for non-linear development
(thousands of parallel branches)

Fully distributed

Able to handle large projects like the Linux kernel
efficiently (speed and data size)

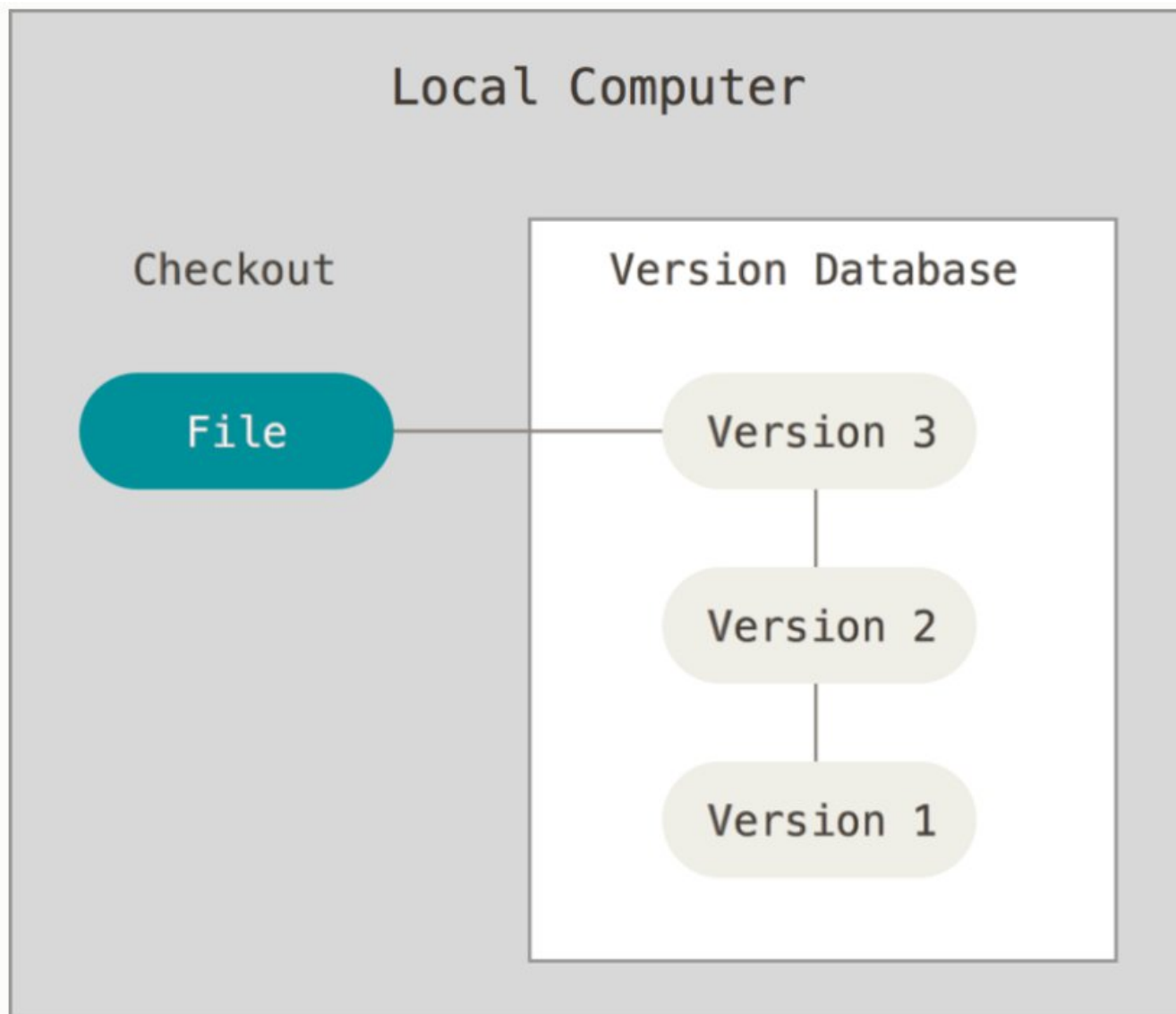
Работает в каталоге (репо, .git)

Изменения в файле

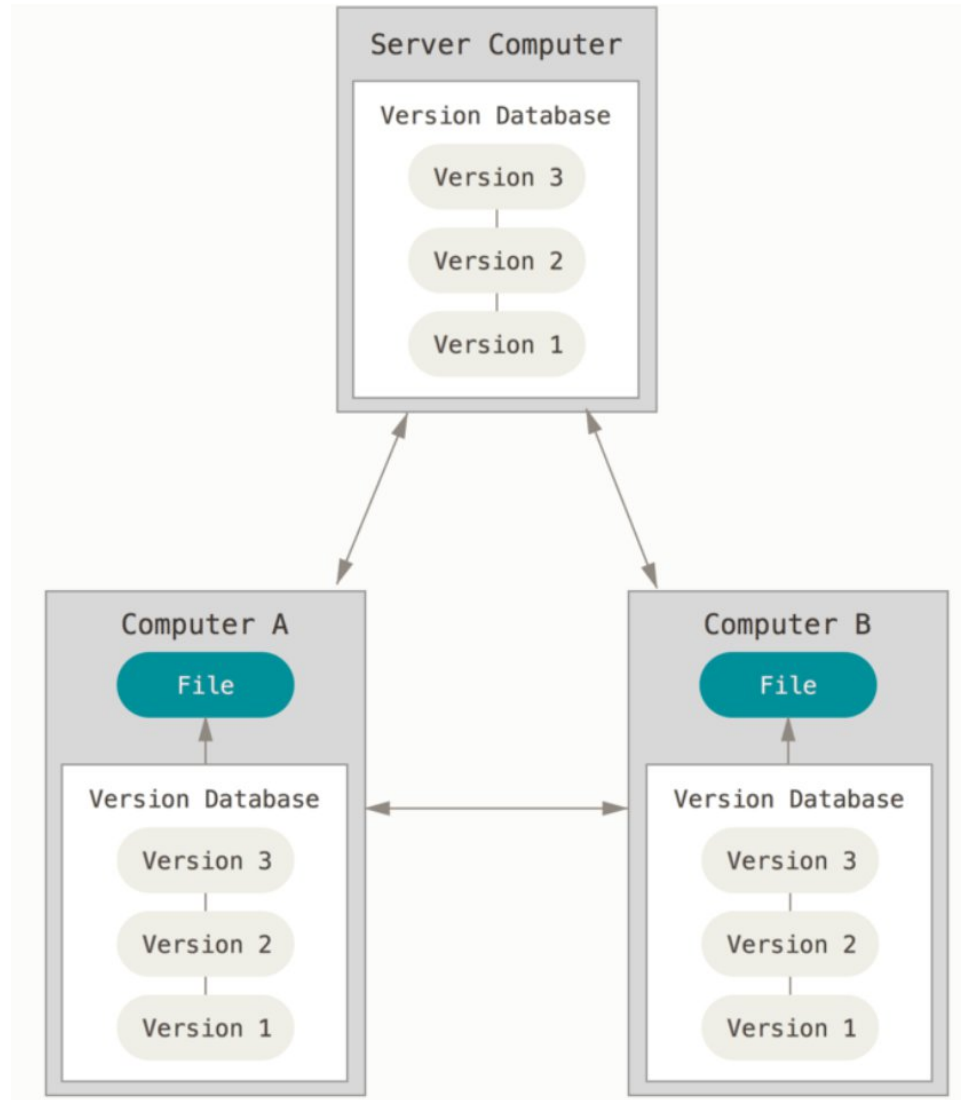
Создание файла

Переименование файла

Стратегии слияния



git



git

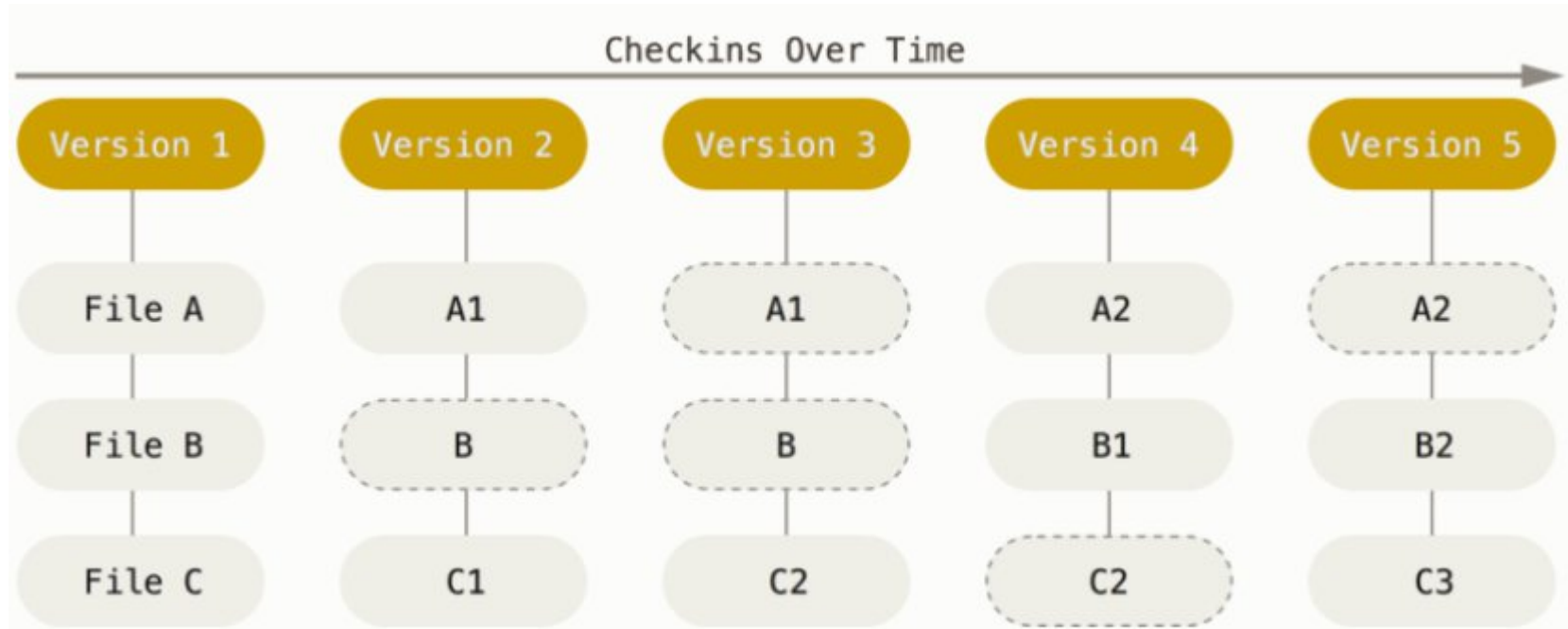
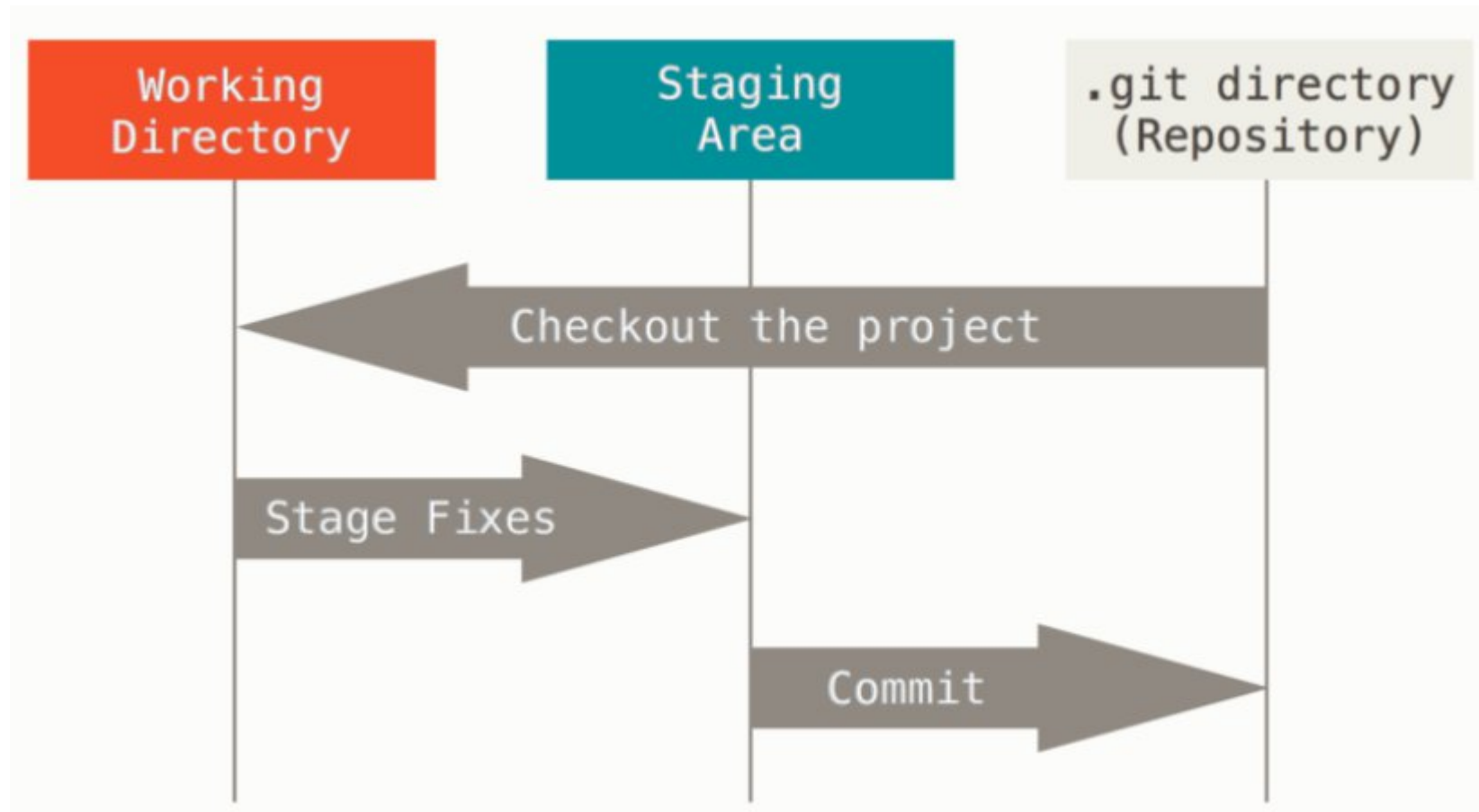


Figure 5. Storing data as snapshots of the project over time

git



Перед началом работы

Проинсталлировать git:

```
sudo apt install git
```

Посмотреть настройки

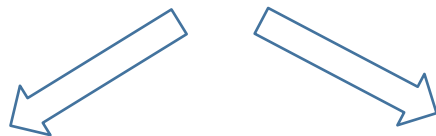
```
git config --list --show-origin
```

Задать обязательные начальные настройки:

```
git config --global user.name "John Doe"
```

```
git config --global user.email johndoe@example.com
```

Начать работать с репозиторием



Локально

Берем любой каталог (даже не пустой)

```
git init
```

```
$ git add *.c
```

```
$ git add README.md
```

```
$ git commit -m 'Initial project version'
```

```
...
```

Клонируем с сервера

```
git clone https://url-to-repo  
cd repo
```

```
$ git status
```

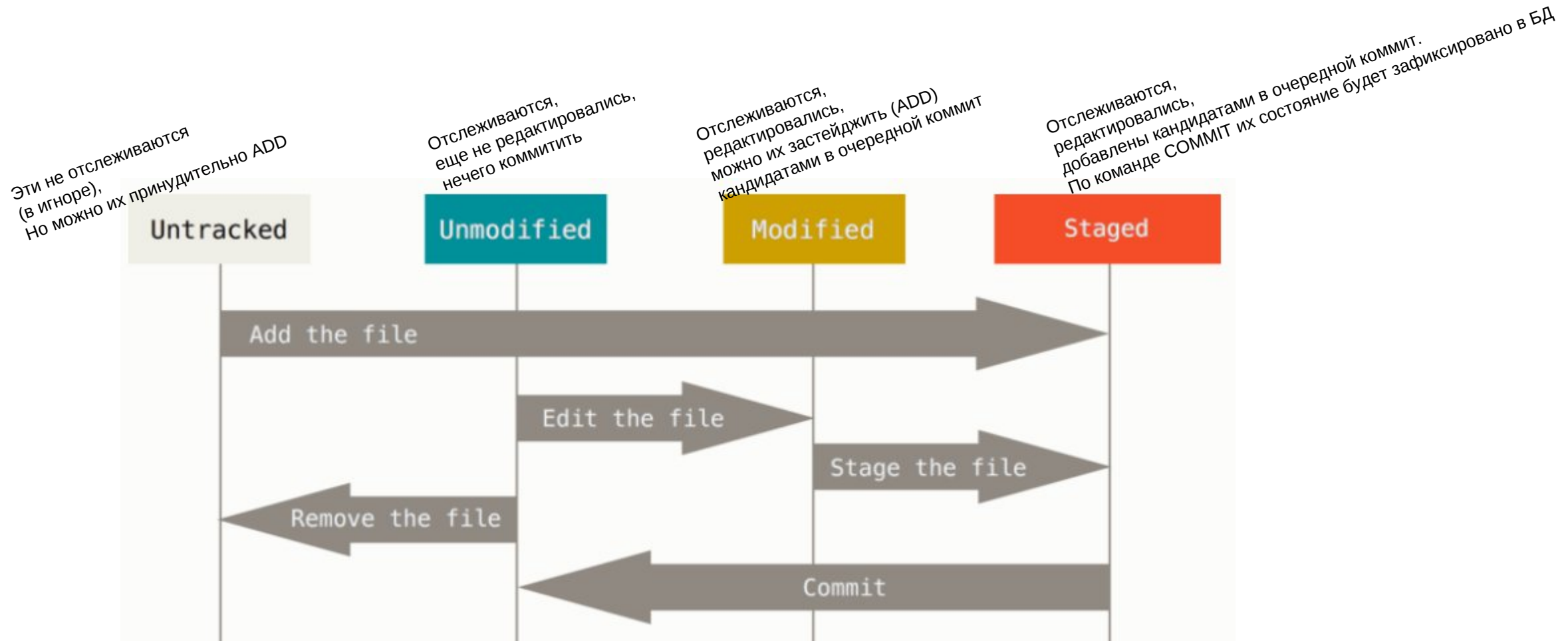
```
On branch master
```

```
Your branch is up-to-date with  
'origin/master'.
```

```
nothing to commit, working tree clean
```

```
$ git checkout xxx
```

Статусы файлов



КОММИТИМ

(как засейвиться и что засейвить)

git status

On branch master

Changes not staged for commit:

(use "git add <file>..." to update what will be committed)

(use "git checkout -- <file>..." to discard changes in working directory)

modified: blabla.py

git add blabla.py

git commit -m «BUG-123: Fixed - rollback feature X»

git push

git log

rm useless.crap

git rm useless.crap (-f)

.gitignore

Журнал

```
git log
git log --stat
git log --oneline
```

```
git checkout a11bef
```

```
git reflog
```

```
$ git log --stat
commit ca82a6dff817ec66f44342007202690a93763949
Author: Scott Chacon <schacon@gee-mail.com>
Date:   Mon Mar 17 21:52:11 2008 -0700

    Change version number

Rakefile | 2 +-
1 file changed, 1 insertion(+), 1 deletion(-)

commit 085bb3bcb608e1e8451d4b2432f8ecbe6306e7e7
Author: Scott Chacon <schacon@gee-mail.com>
Date:   Sat Mar 15 16:40:33 2008 -0700

    Remove unnecessary test

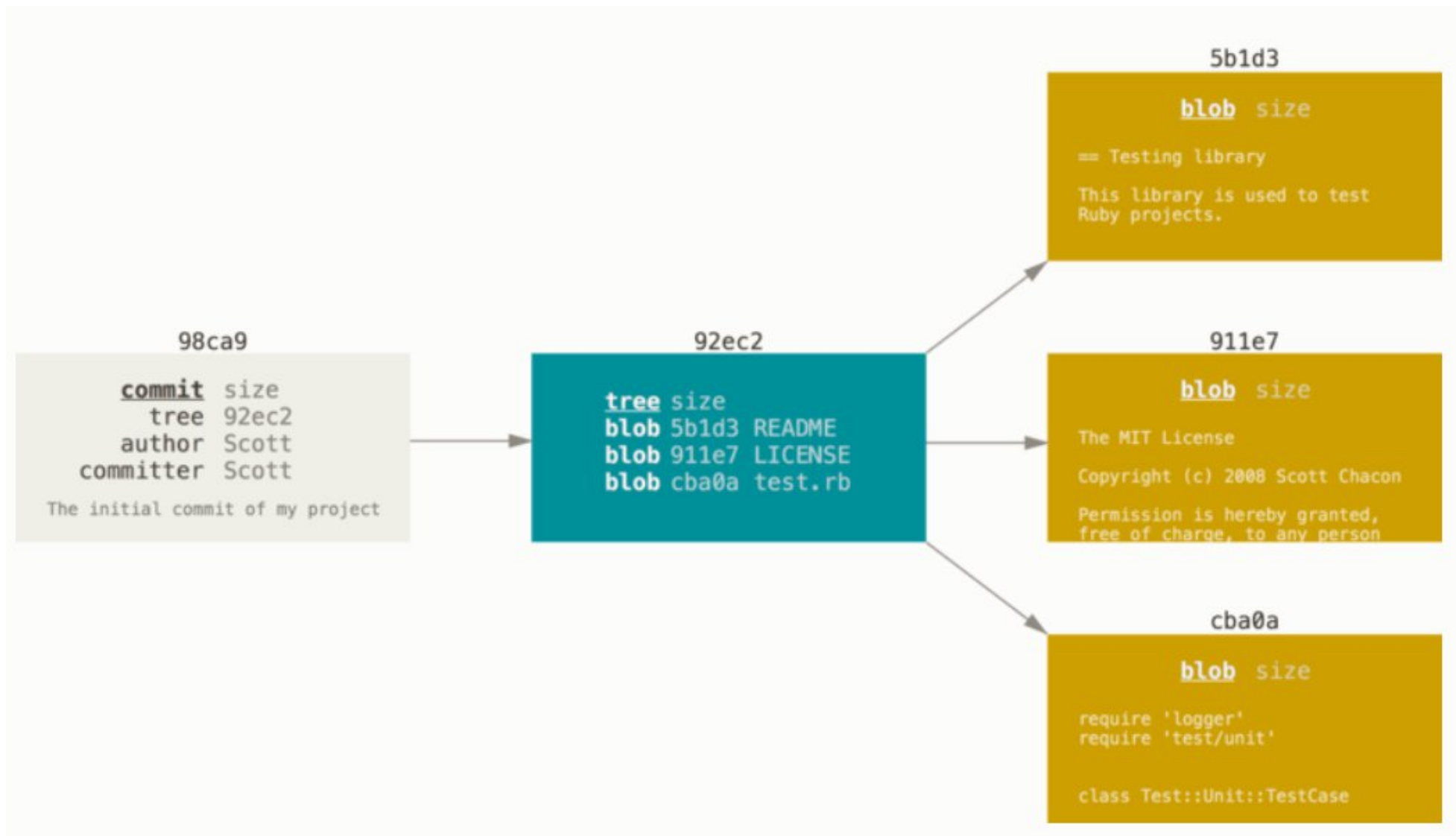
lib/simplegit.rb | 5 -----
1 file changed, 5 deletions(-)

commit a11bef06a3f659402fe7563abf99ad00de2209e6
Author: Scott Chacon <schacon@gee-mail.com>
Date:   Sat Mar 15 10:31:28 2008 -0700

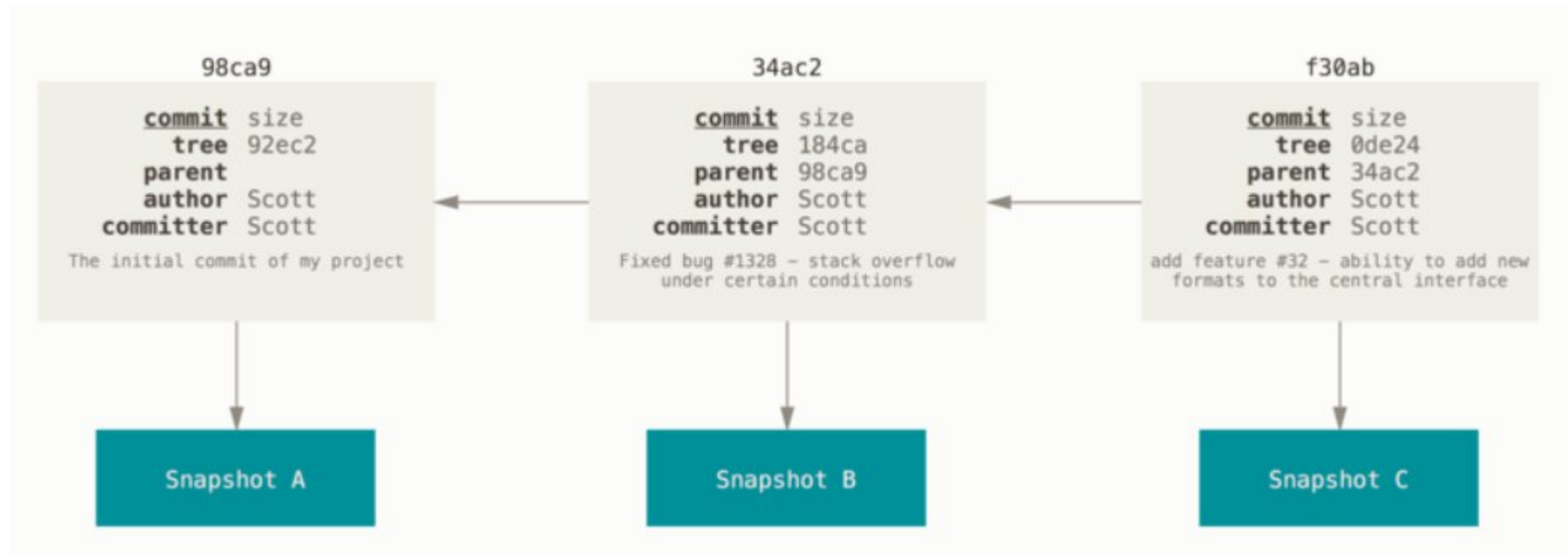
    Initial commit

README          | 6 ++++++
Rakefile        | 23 ++++++++++++++++++++++
lib/simplegit.rb | 25 ++++++++++++++++++++++
3 files changed, 54 insertions(+)
```


Что внутри «коммита»

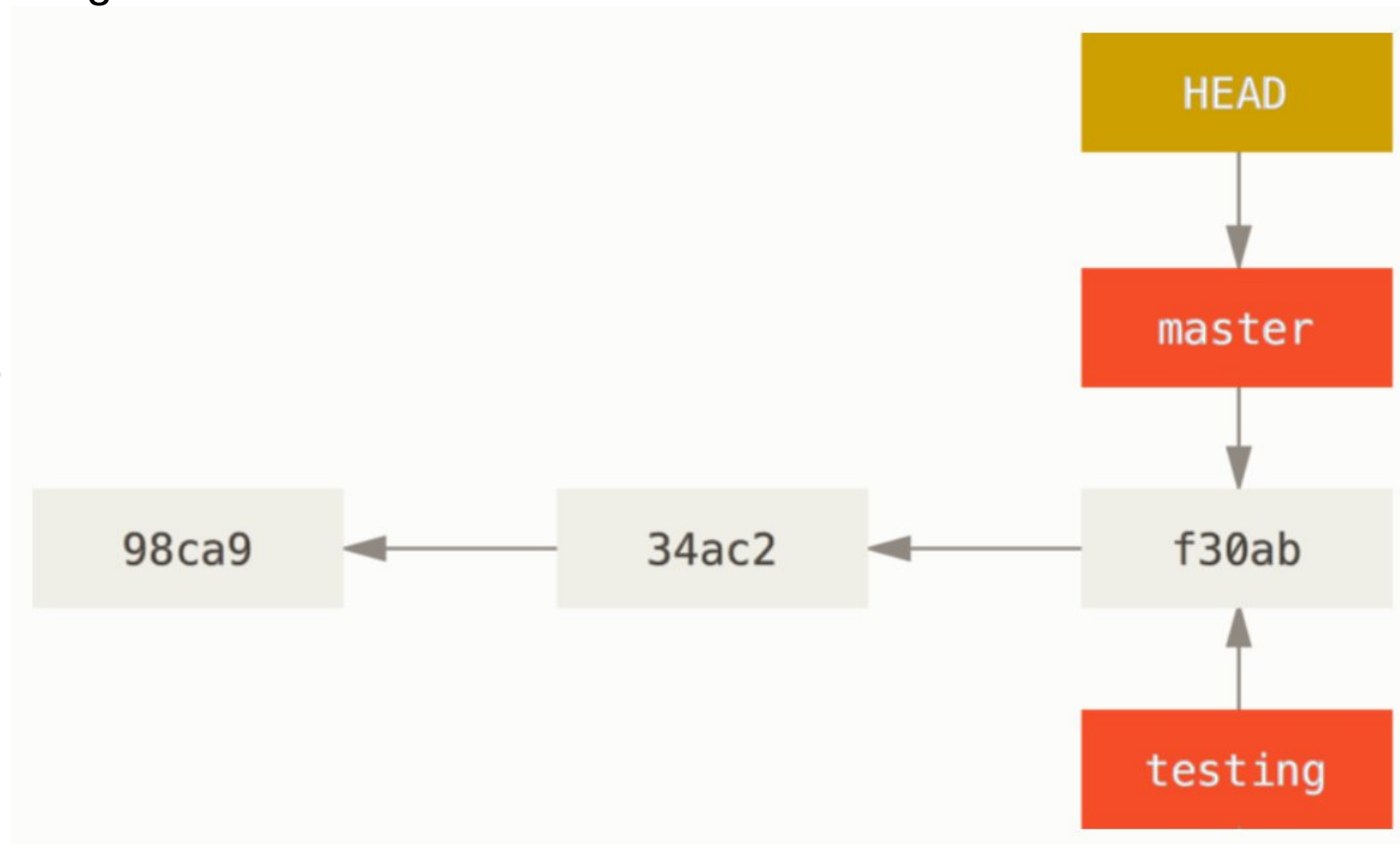


Цепочка коммитов (ветка)

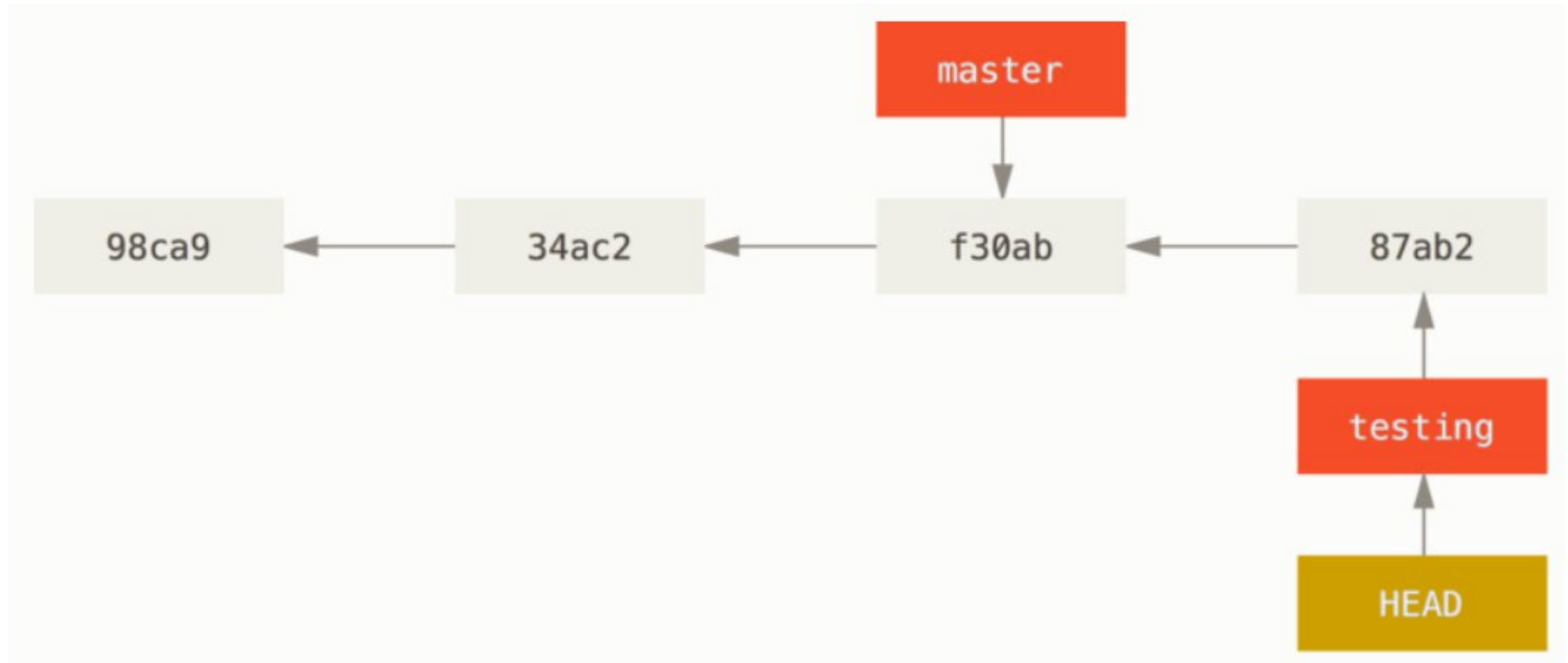


git checkout master
git branch testing

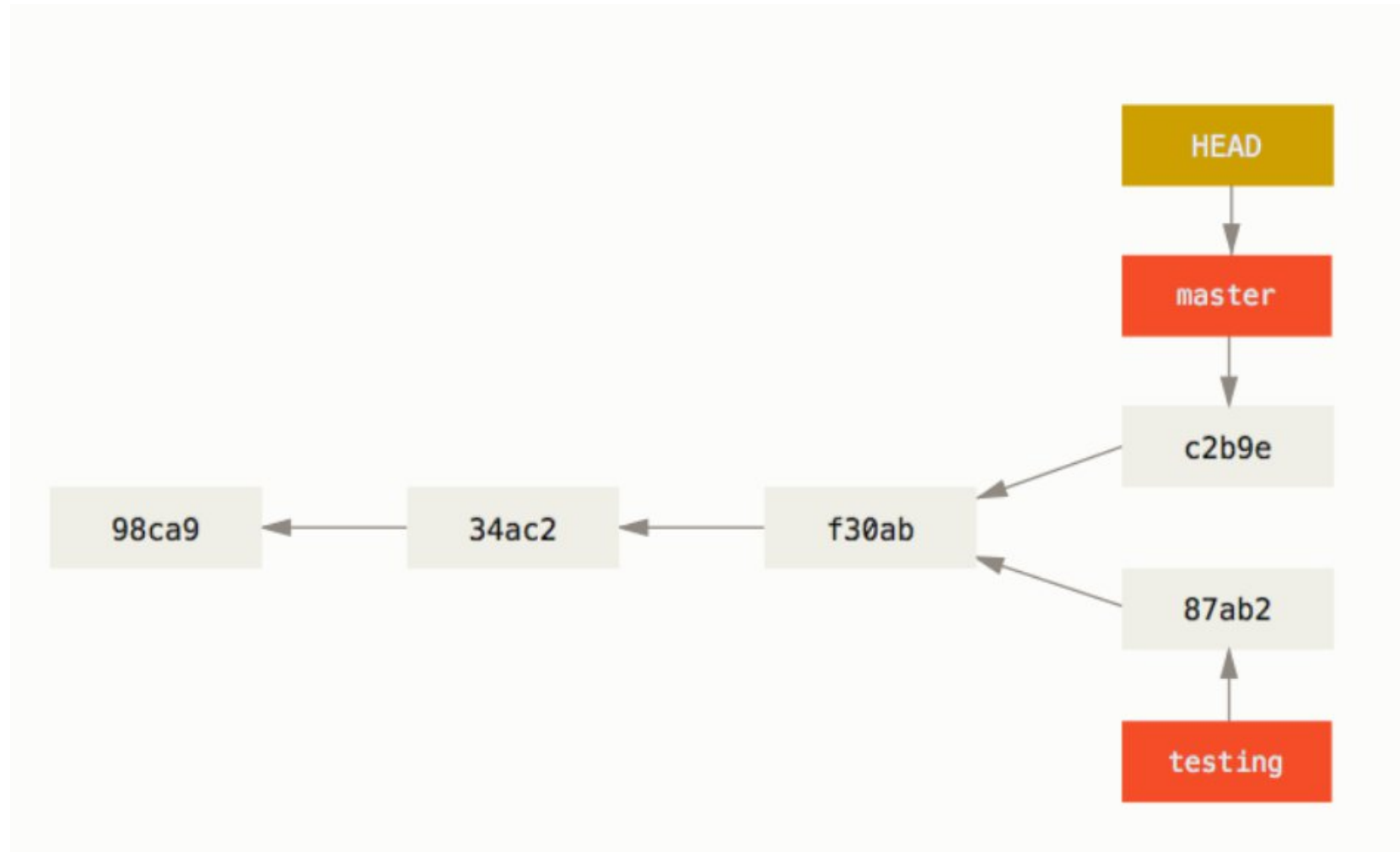
master/main – просто дефолт,
Можно менять имя



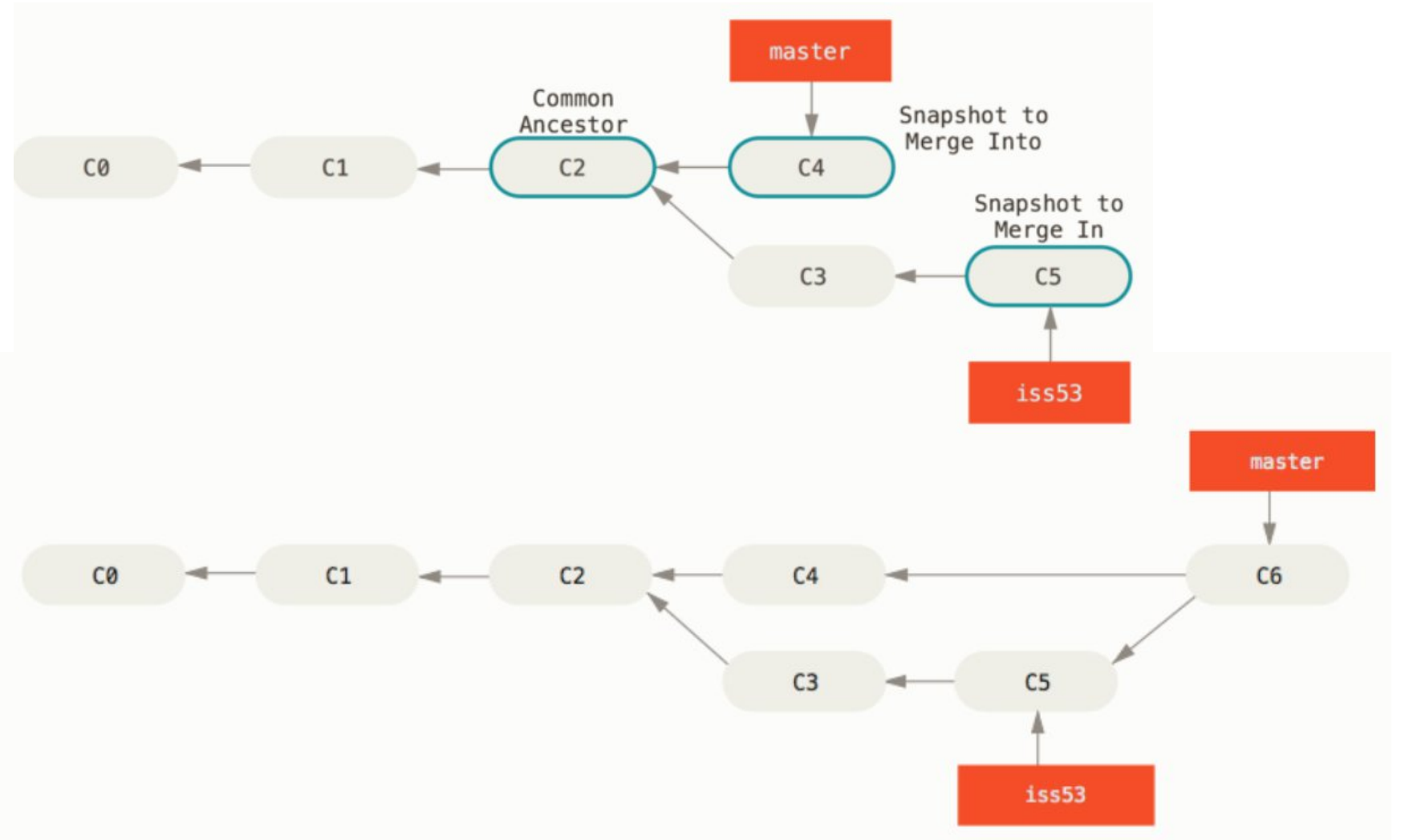
git checkout testing
git commit



git checkout master
git commit

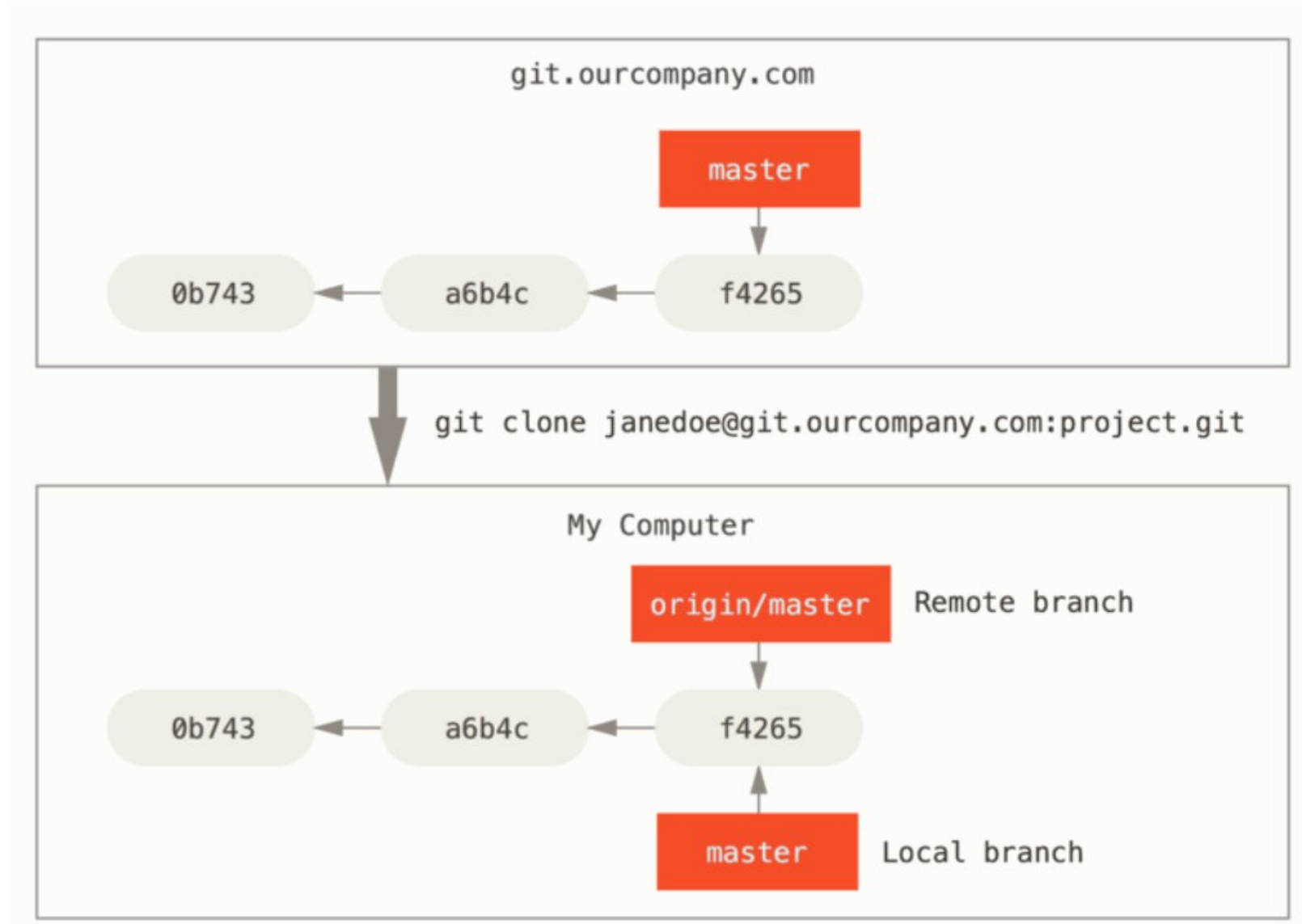


git checkout master
git merge iss53

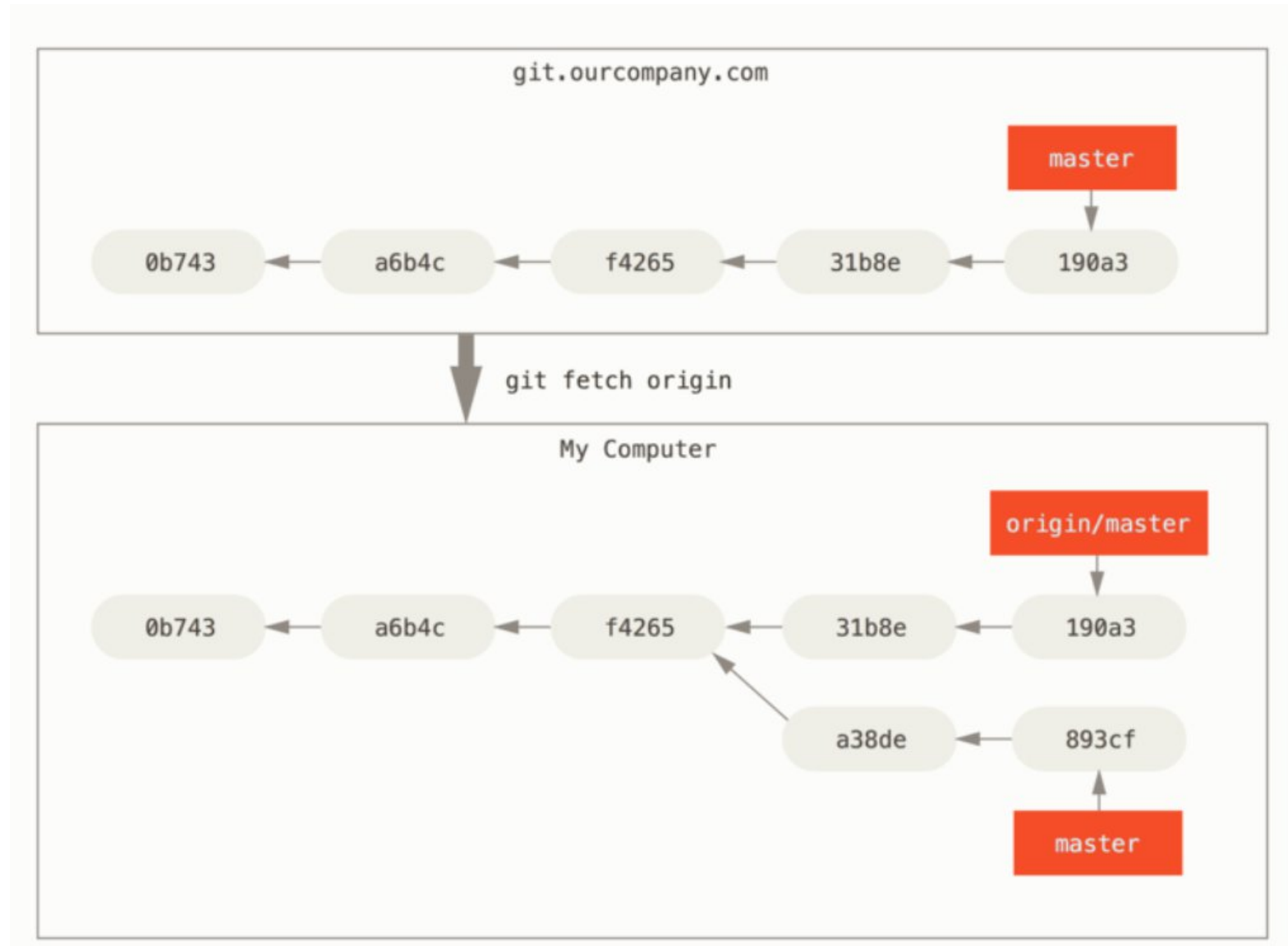


Удаленные ветки (remote branches)

Origin – просто дефолтное имя,
можно менять



git fetch



git pull

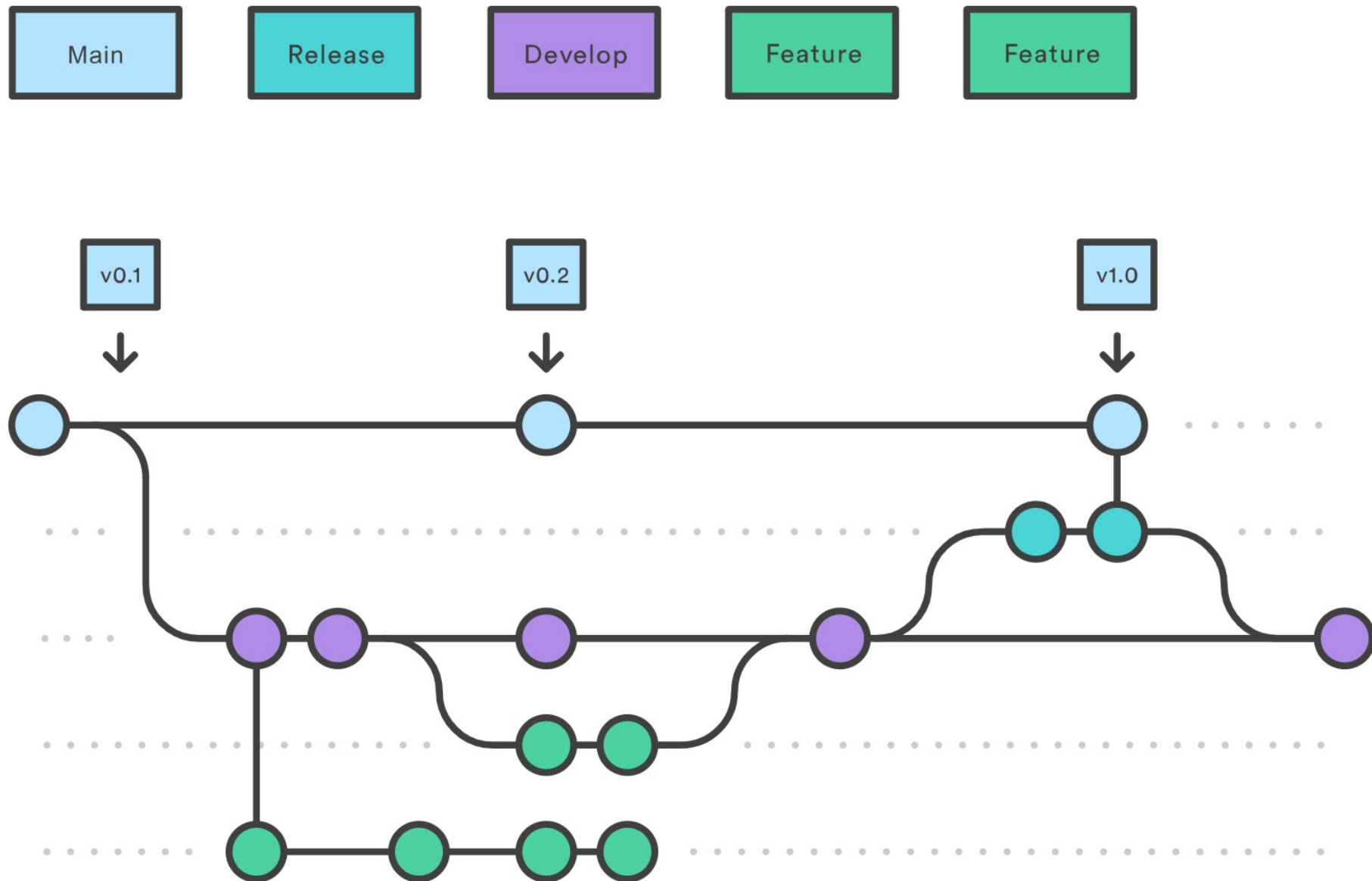
git pull = git fetch + git merge

git push

! Если пушим ветку, которая уже есть на сервере (отслеживается у нас)
git push

! Если пушим новую ветку (включим отслеживание)
git push -u origin my-new-branch

Стратегия GitFlow



GitHub/GitLab

Branch protection

Merge request

Pull request

Порядок работы

```
git clone https://url-to-repo  
git checkout topic
```

```
git status  
git pull  
<РАБОТАЕМ>
```

```
git status  
git add -u  
git commit -m «update some»  
git push
```

```
git checkout master  
git pull
```

```
Git checkout topic  
Git merge master  
<ПРАВИМ КОНФЛИКТЫ>  
Git commit  
git checkout master  
Git merge topic  
Git push
```

Справка

<https://ohshitgit.com/>

Fuck this noise, I give up.

```
cd ..  
sudo rm -r fucking-git-repo-dir  
git clone https://some.github.url/fucking-git-repo-dir.git  
cd fucking-git-repo-dir
```

Thanks to Eric V. for this one. All complaints about the use of `sudo` in this joke can be directed to him.

П03. git

1. Зарегистрироваться на Github
2. Добавить SSH ключ или активировать 2FA

3. Задача:

- клонировать репо <https://github.com/vparanoid/xxxxxxxxxx>
- создать в репо файл с любым именем
- в файл list дописать своё имя и имя созданного вами файла
- сделать коммит
- запушить изменения (по пути решив возникающие конфликты)
- получить права коллаборатора на репо

Использовать

git clone

git pull

git add/commit

git push

П03. Кейсы

Permission denied

- Не добавлен SSH ключ в свой профиль – добавить
- Не добавлен пользователь в коллабораторы – добавить
- Новая win/mingw
 - Попробовать добавить ключ в ssh-agent
 - <https://docs.github.com/en/authentication/troubleshooting-ssh/error-permission-denied-publickey>
 - <https://docs.github.com/en/authentication/connecting-to-github-with-ssh/generating-a-new-ssh-key-and-adding-it-to-the-ssh-agent>
- Делать лабу на линуксе (добавить в свой гитхаб профиль SSH ключ из линукса)
- Поменять способ работы с репо с SSH (git) на HTTPS
 - `git remote set-url origin https://repo`
 - `git remote set-url origin git@github.com:vparanoid/itech2024-2.git`

П03. Конфликт

- Вы правите локальный файл, пытаетесь запушить, а там уже есть чужие изменения
 - Сделать `git pull`
 - Починить конфликты, если есть
 - `git add file.x`
 - `git commit -m «Исправил конфликт»`
 - `git push`