

Изменить дополнительную информацию

Разработка на языке Python

GIT



Задача

Команде требуется сделать проект который

1. Делают несколько человек
2. Имел бы девелоперскую версию и релизную без лишней путаницы
3. Можно было бы откатить до более ранних версий
4. Имел бы несколько параллельно решаемых задач
5. С легкой отслеживаемостью изменений в версиях

Решения?

Иметь на своем компьютере по отдельной папке на каждую версию проекта

Каждую новую версию мы отдаем коллегам по флешке, а они свои изменения также отдают нам на флешке.

На чьем компьютере актуальная версия?

Иметь центральном сервере лежит актуальная версия проекта. Все в команде перекидывают туда обновления, а один человек сшивает в новую версию

Если менялся один файл двумя людьми, приходится сравнивать код всех новых версий этого файла и сшивать их в один.

Точно ли были учтены все изменения?

Сохраняем не проект, а изменения проекта сделанные каждый человеком и отправляем их на сервер. С сервера получаем информацию о всех изменениях, сделанных командой.

А если сервер упадет и не встанет вместе со всей работой?

CVS

Система управления версиями - программа, способная запоминать каждое изменение или состоянии информации. Также возможны откат к прошлым версиям, определять кто и когда делал изменения, строить новые версии от более старых версий проекта

Также возможны отправка репозиторий всей информации на проекте и получения обновлений из удаленного репозитория

Решения





Разные CVS



git



Git



- Создана Линусом Торвальдсом
- Создана для нужд разработчиков ядра Linux
- Первая версия вышла в 2005г

Git

- Наиболее быстрая работа с локальным репозиторием
- Удобная система ветвления
- Полный функционал для работы с коммитами
- Возможность исправить любую ошибку работы с Git
- Огромный функционал для нахождения разности между ветками
- Легкое решение конфликтов между ветками
- Огромное количество учебной литературы и туториалов
- Наиболее популярная система среди разработчиков(>75%)
- Бесплатна и открыта
- Много вариантов работы с удаленном репозиториием

Не путать с
хостингами
репозиториев!

ХОСТИНГИ

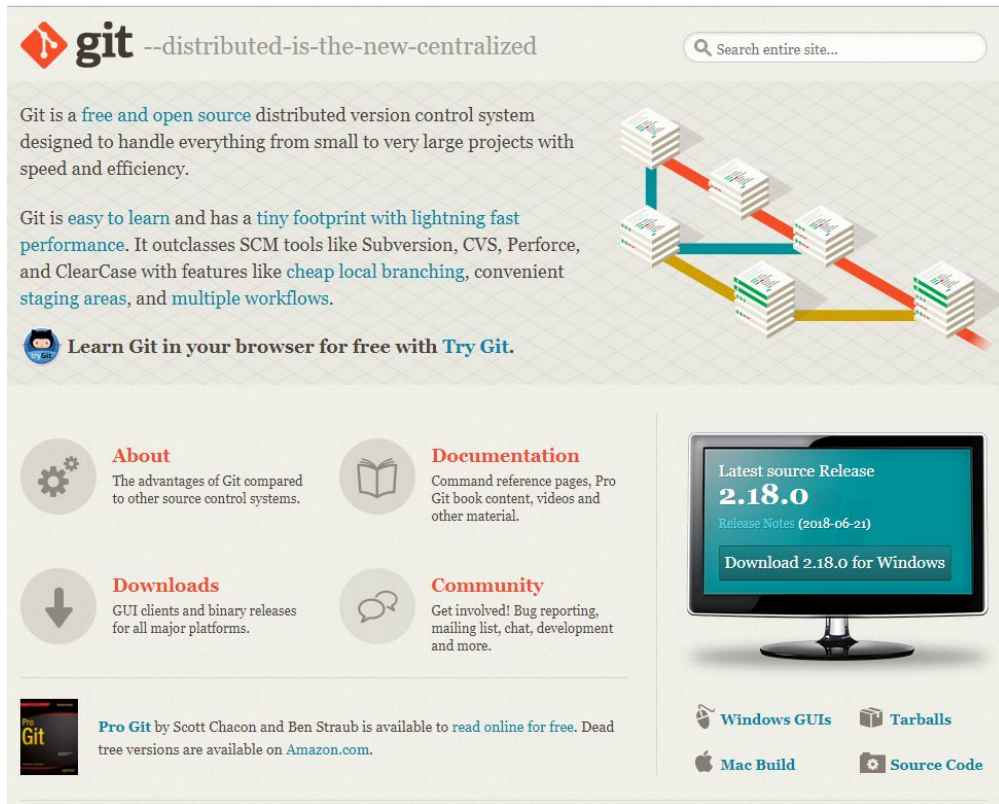


GitLab

Установка Git

Бесплатно установить на любую систему можно с официального сайта git

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




The screenshot shows the Git website homepage. At the top, the Git logo is followed by the tagline "--distributed-is-the-new-centralized". A search bar is in the top right. Below the header, a paragraph describes Git as a free and open source distributed version control system. To the right is a diagram showing a distributed version control system with multiple repositories connected by lines. Below this, another paragraph describes Git as easy to learn and having a tiny footprint with lightning fast performance. A link to "Learn Git in your browser for free with Try Git" is provided. The main content area is divided into four sections: "About" (advantages of Git), "Documentation" (command reference, Pro Git book), "Downloads" (GUI clients, binary releases), and "Community" (bug reporting, mailing list). On the right, a monitor displays the "Latest source Release 2.18.0" and a button to "Download 2.18.0 for Windows". At the bottom, there are links for "Windows GUIs", "Tarballs", "Mac Build", and "Source Code".

git --distributed-is-the-new-centralized

Search entire site...

Git is a **free and open source** distributed version control system designed to handle everything from small to very large projects with speed and efficiency.

Git is **easy to learn** and has a **tiny footprint with lightning fast performance**. It outclasses SCM tools like Subversion, CVS, Perforce, and ClearCase with features like **cheap local branching**, convenient staging areas, and **multiple workflows**.

 **Learn Git in your browser for free with Try Git.**

About
The advantages of Git compared to other source control systems.





Documentation
Command reference pages, Pro Git book content, videos and other material.

Downloads
GUI clients and binary releases for all major platforms.

Community
Get involved! Bug reporting, mailing list, chat, development and more.

Latest source Release
2.18.0
Release Notes (2018-06-21)
Download 2.18.0 for Windows

Pro Git by Scott Chacon and Ben Straub is available to [read online for free](#). Dead tree versions are available on [Amazon.com](#).

 **Windows GUIs**  **Tarballs**
 **Mac Build**  **Source Code**

Начало работы

Перейдя в нужную папку можно инициализировать работу двумя способами

```
cd /home/usr/..../wokplace
```

Можно создать из этой папки чистый локальный репозиторий

```
git init
```

Или скопировать в эту папку уже существующий с удаленного сервера

```
git clone URL
```

Отслеживание файлов

Для того, чтобы начать работать с файлом, сначала нужно начать его отслеживать.

Сделать это можно с помощью команды **add**

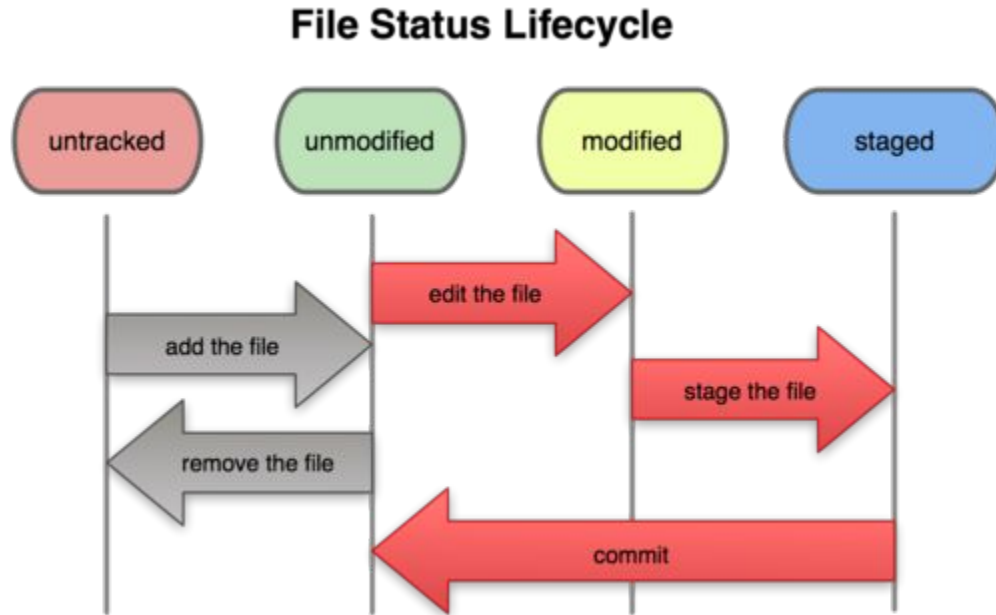
```
git add filename
```

После этого система знает, что <filename> часть версии проекта

Проверить отслеживаемые файлы можно с помощью команды **status**

```
git status
```

Untracked vs unstaged



Коммиты

Коммит - это изменения в проекте отслеживаемых файлов, относительно прошлого изменения

Коммит - основная рабочая единица гита

Для того, чтобы сделать коммит достаточно команды

```
git commit
```


Ветвления

Основная фишка GIT - ветвления. При изначальном создании репозитория GIT создаются изначальный коммит и главная ветка **master**

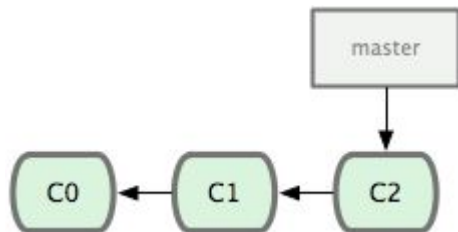
Ветвить можно командами **branch** и **checkout** с флагом **-b**

```
git checkout -b NAME  
git branch NAME
```

Переключать ветки можно с помощью команды **checkout**

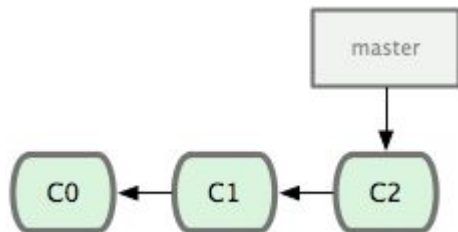
```
git checkout NAME
```

Ветвления

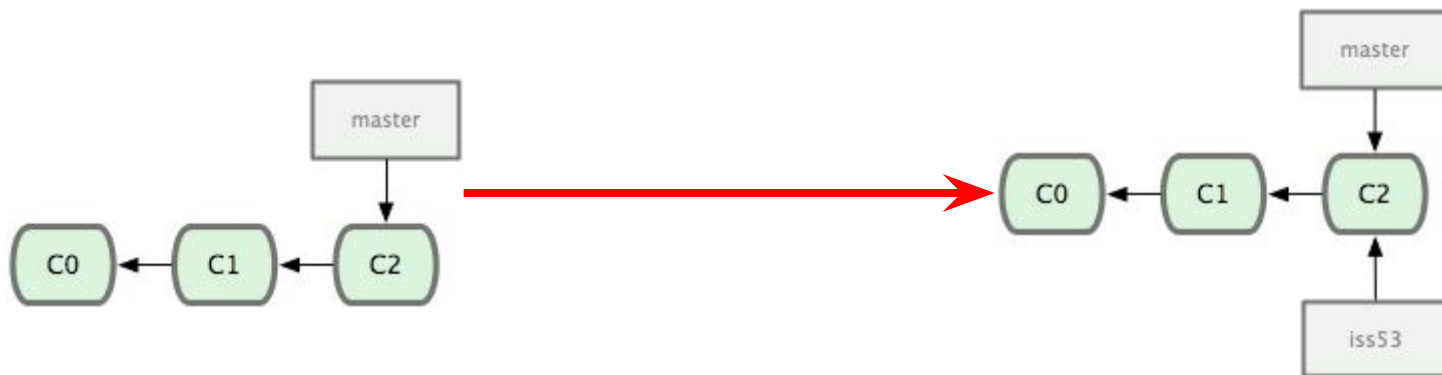


Ветвления

```
git checkout -b iss53
```



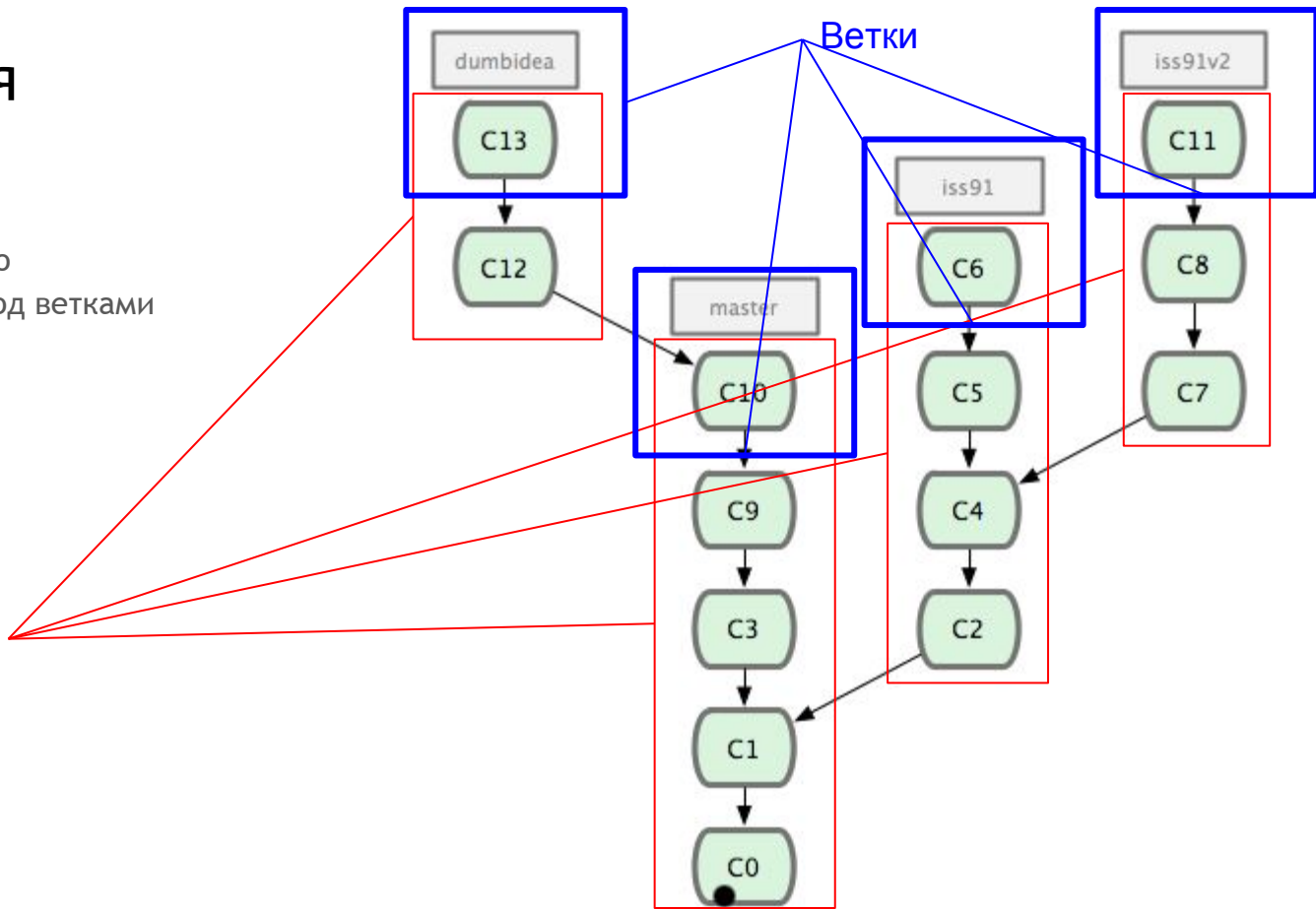
Ветвления



Ветвления

Стоит понимать, что подразумевается под ветками

НЕ ВЕТКИ!



Переключаться
между ветками
можно в любой
момент не теряя
своей работы!

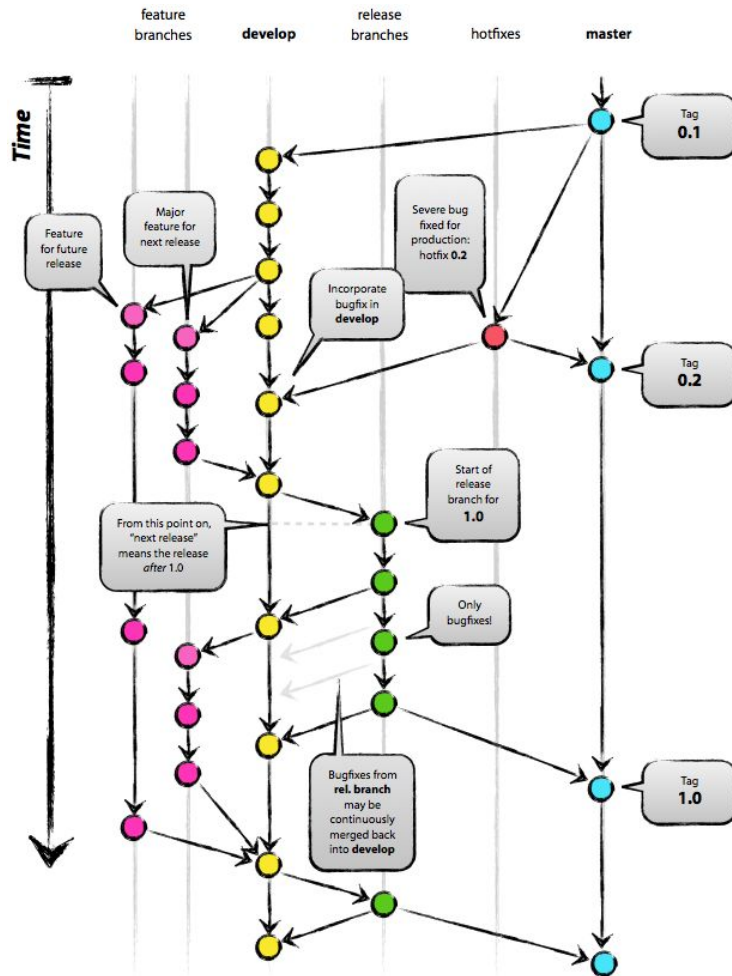
Граф	Рев	Ветка	Описание
	3153+	Live	★ Рабочий каталог ★
	3160	Dev2	Dev2 tip Слияние с default
	3159	default	default More readable error messages by email.
	3158	default	Create AdminTable->to->xlsx driver.
	3157	Dev2	Пофиксил баг с открытием вакансий, после ух
	3156	3d-live	3d-live Слияние с 3d
	3155	3d	3d Дефолтный скейл равен 1
	3154	3d	Анимация с помощью TweenLite
	3153	Live	Live Слить
	3152	Dev2	Текст
	3151	Dev2	Шеринг
	3150	3d	Правка размера текста в слепую
	3149	3d	Правка прилета шаров
	3148	Live	Слить
	3147	Dev2	Слить
	3146	Dev2	...

Ветвления

В больших проектах всегда существует регламент на правильное ветвление

Обычно существует ветка master в которой лежат только релизы, ветка develop, в которой происходит работа, и много веток feature, где работают разные команды

За большой информацией гуглите
Git flow !



Слияние веток

Существуют два способа соединить два коммита - **merge** и **rebase**

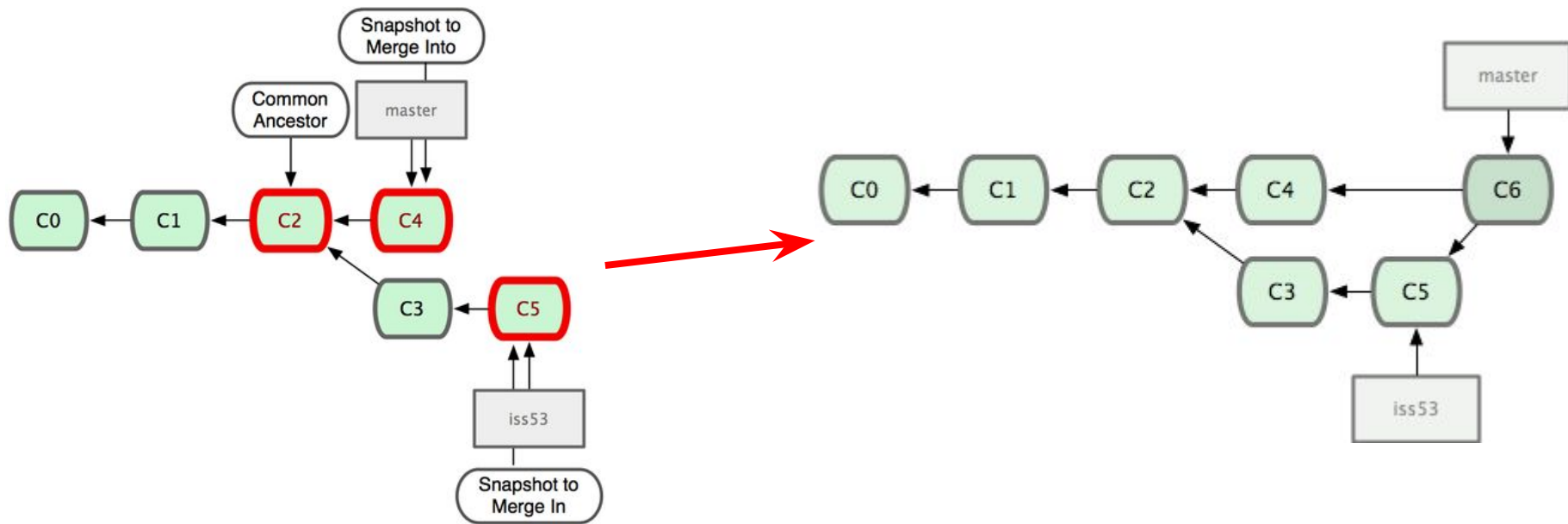
Слияние - добавление всех изменений с одной ветки на другую

```
git merge NAME
```

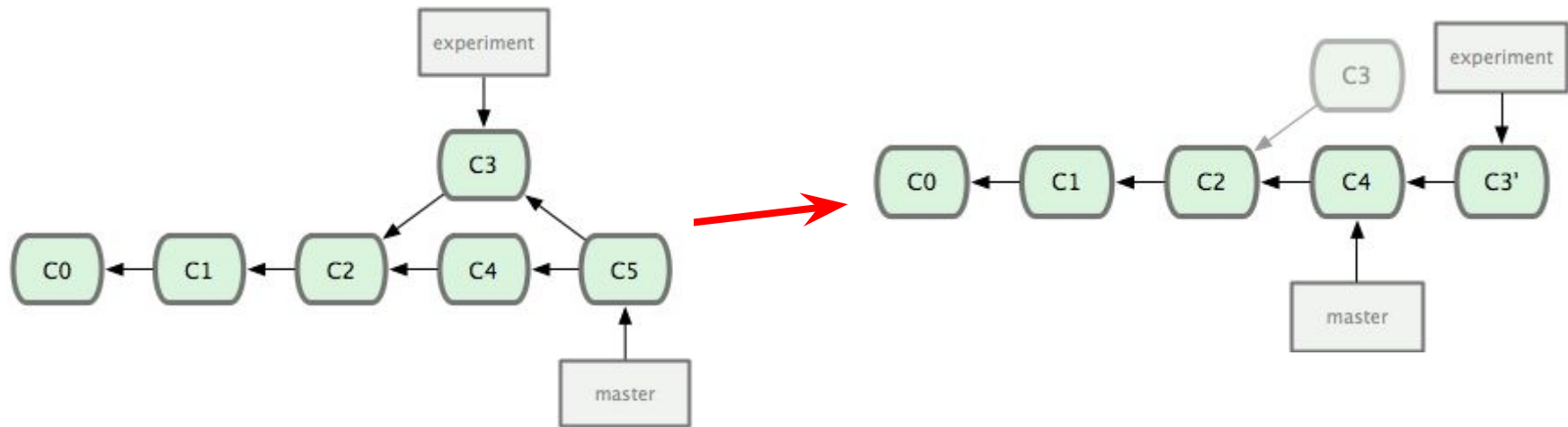
Перемещение - изменение первоначального коммита, с которого когда то была создана ветка

```
git rebase NAME
```

Merge



Rebase



Rebase vs Merge

Слияние веток (merge)

Требуется находиться на ветке в которую вливают и вливать относительно второй

```
git checkout feature  
git merge master
```

Перенос одной ветки на другую (rebase)

Требуется находиться на той ветке которую перемещают и перемешать относительно второй

```
git checkout master  
git rebase feature
```

Никогда не
перемещайте(rebase)
ветки, которые
вытащили из
удаленного
репозитория !!!!!!!!

Работа с удаленным репозиторием

Работа никогда не ограничивается только в локальном репозитории. Не только постоянно приходится обмениваться наработками с командой, но и подгружать изменения в основных ветках и перебазировать свои изменения, чтобы избегать конфликтов в проекте.

Требуется научиться как получать обновления из центрального репозитория, но и отправлять свои ветки

Получение новых веток

Получить данные из удаленного репозитория можно командой fetch

```
git fetch origin
```

Где origin - дефолтное название удаленного репозитория, откуда был сделан изначальный git clone

Обновление веток

С помощью команды `git pull` можно получить из удаленного репозитория информацию и слить ее в ту ветку над которой вы сейчас работаете

```
git fetch pull
```

Выполнив эту команду Git сделает `fetch` с `origin` и попытается слить (`merge`) вашу ветку над которой вы работаете, и сливает ее с той, которая придет после `fetch`

Отправка ветки

Для того чтобы отправить свою ветку на удаленный сервер существует команда `push`

```
git push origin NAME
```

Это команда позволит нам отправить нашу ветку на удаленный репозиторий, которую увидят все, после очередной команды **fetch**

Внесения вклада в проект

Отправить свою ветку на удаленный репозиторий - недостаточно! В первую очередь вы хотите, чтобы ваша ветка, над которой вы работали какое-то время стала частью проекта (например частью ветки develop)

Для этого вы делаете push своей ветки, после чего делаете pull request на сервере

commits

3 branches

0 releases

1 contributor

Branch: master

New pull request

Create new file

Upload files

Find file

Clone or download

Cerber47 Add files via upload

Latest commit 7415408 3 days ago

ClassExamples Adding new material

a month ago

Hometask New hot fixes

28 days ago

Materials Add files via upload

3 days ago

README.md New read me

28 days ago

README.md



Внесения вклада в проект

1. Создаете ветку (feature)
2. Работаете над ней
3. Сквишите (если этого требует регламент)
4. Проверяете на наличие конфликтов с той веткой, в которую хотите влить (develop ветка например)
5. Пушите (push)
6. Делаете pull request
7. Обсуждаете с админом репозитория косяки и промахи
8. Если получаете отказ - идете дорабатывать, если получаете определенное количество согласий - ветка сливается и удаляется