

INTRODUCTION TO GIT

IHAR PRYSHCHEPAU

OCTOBER, 2019

iTechArt



- ❑ Система контроля версий (VCS)
- ❑ Git

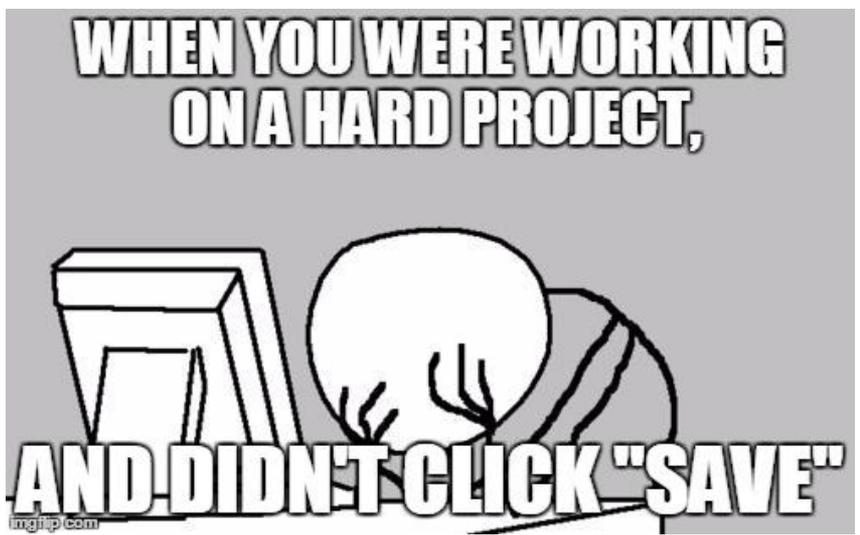
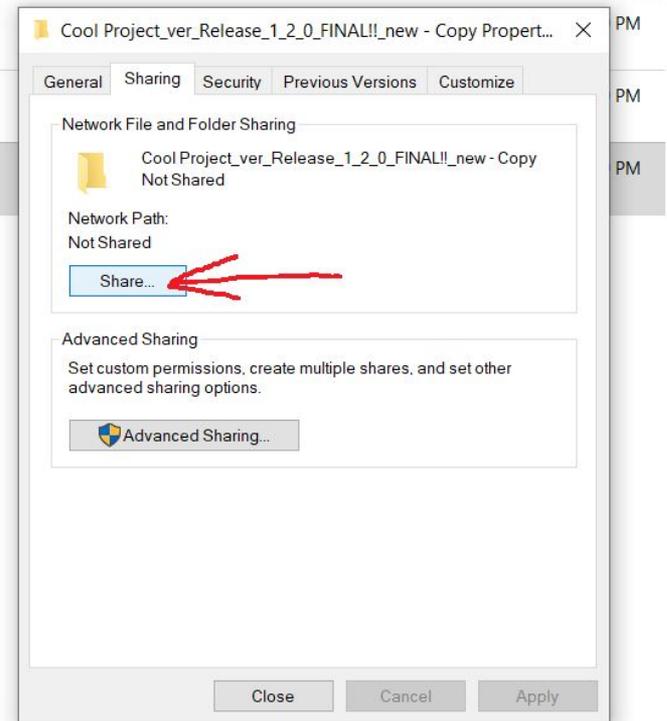
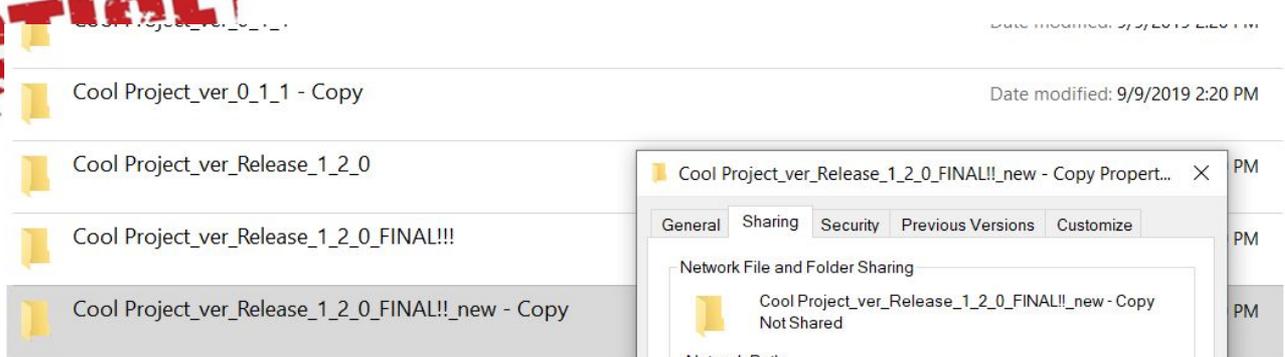
УРОВЕНЬ 1. АВТОНОМНАЯ РАБОТА

BAD PRACTICE

- ✦ Cool Project_ver_0_1_1
- ✦ Cool Project_ver_0_1_1 - Copy
- ✦ Cool Project_ver_Release_1_2_0
- ✦ Cool Project_ver_Release_1_2_0_FINAL!!!
- Cool Project_ver_Release_1_2_0_FINAL!!!_new - Copy

УРОВЕНЬ 2. NETWORK SHARING

BAD PRACTICE!



УРОВЕНЬ 3. CLOUD



Name	Date Modified
Sample File	Today at 10:54
Sample File (hanz-mbp's conflicted copy 2018-10-12)	Today at 10:54

Backup and Sync from Google

 **Continue merge?**

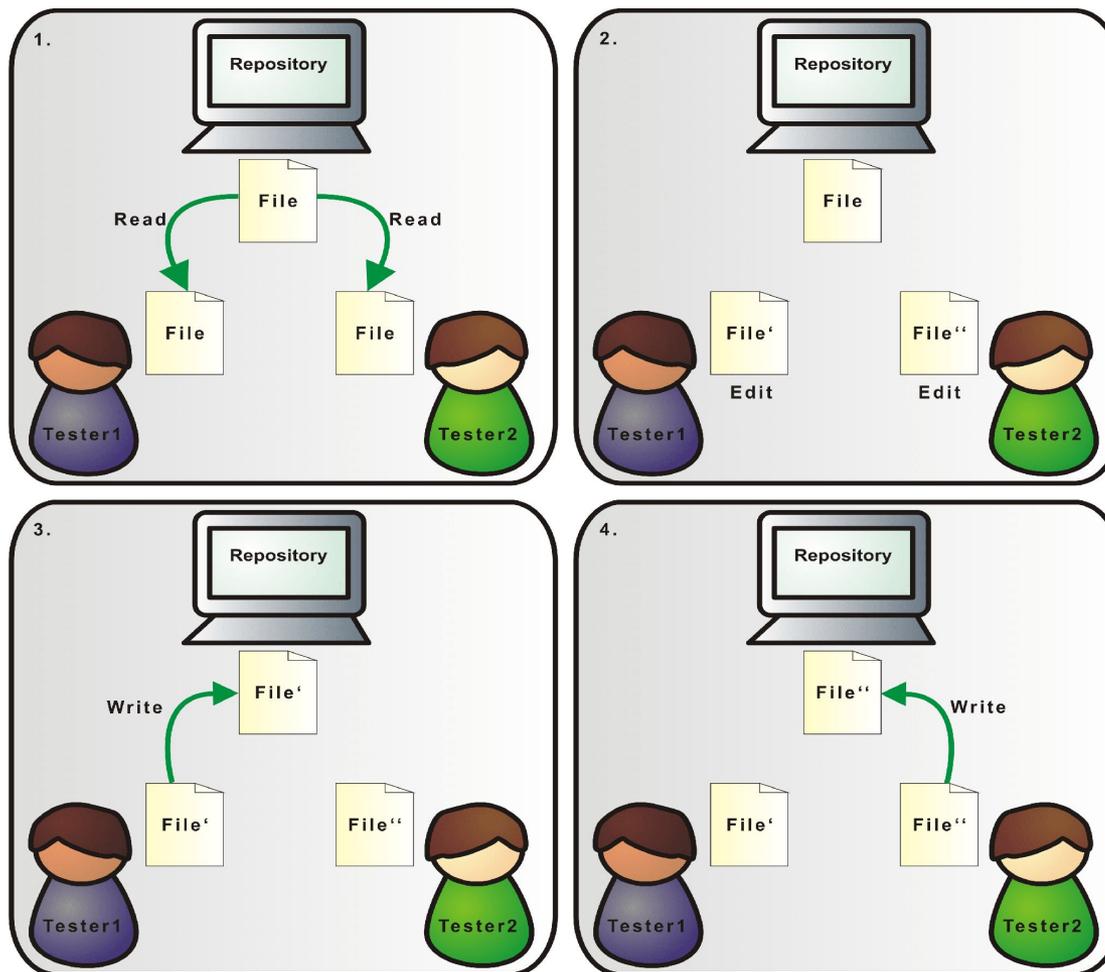
815 items will sync
125 items conflict and will be stored as separate copies

CANCEL CONTINUE

Sync conflicts

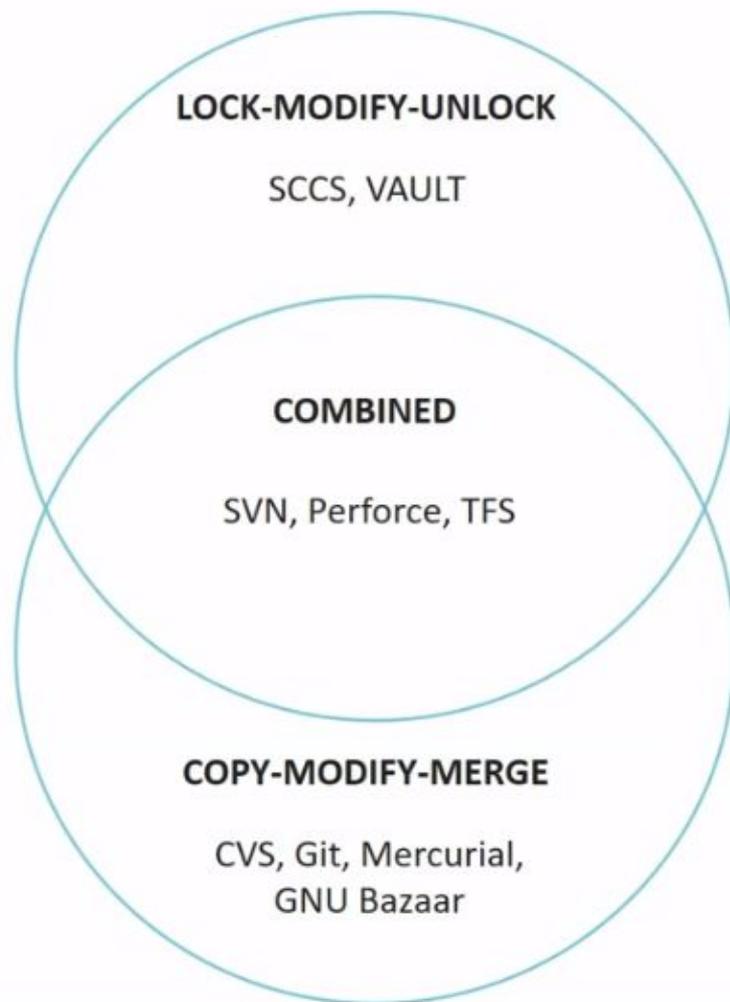
- Let me choose to merge changes or keep both copies
- Always keep both copies (rename the copy on this computer)

ГЛАВНАЯ ПРОБЛЕМА КОНТРОЛЯ ВЕРСИЙ

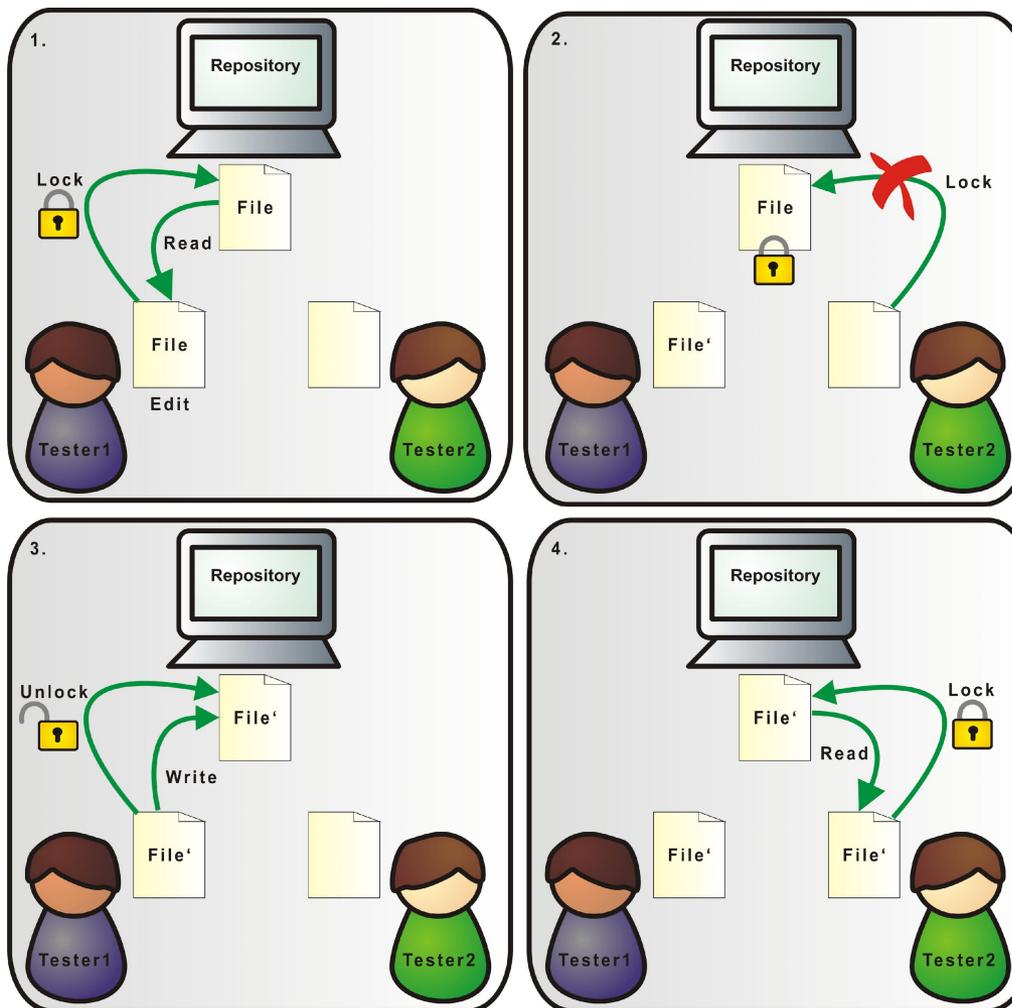


- ✓ Back-up & restore
- ✓ Синхронизация командной работы
- ✓ Отмена изменений
- ✓ Отслеживание изменений и их авторов
- ✓ Sandboxing
- ✓ Возможность работы с ветками

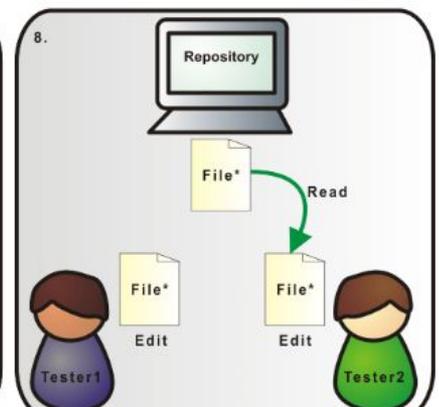
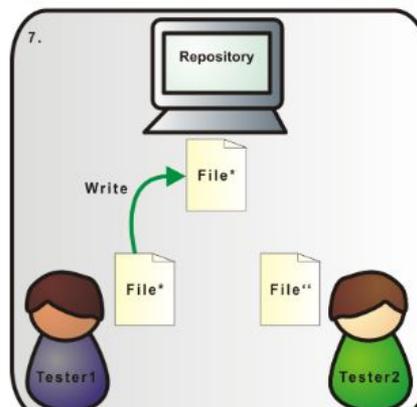
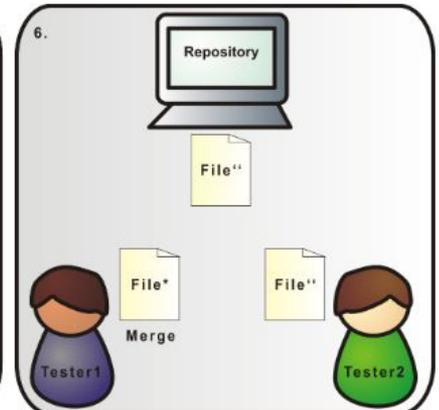
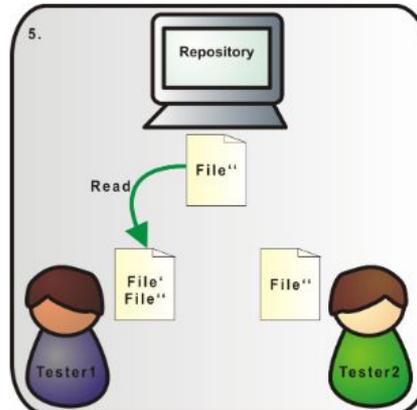
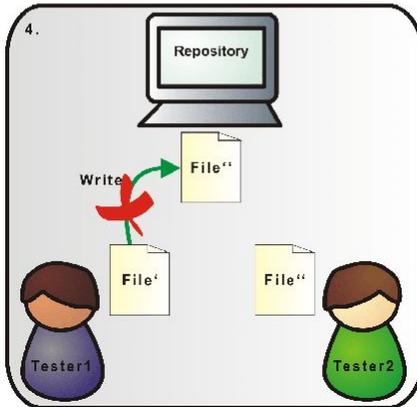
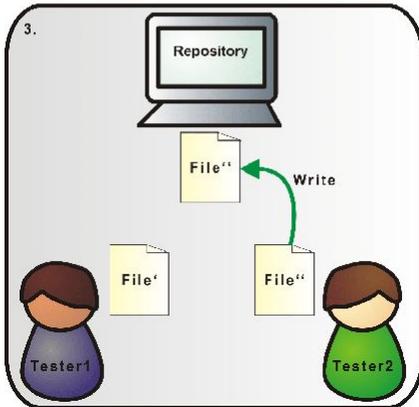
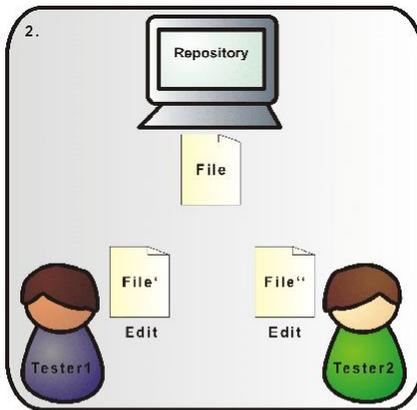
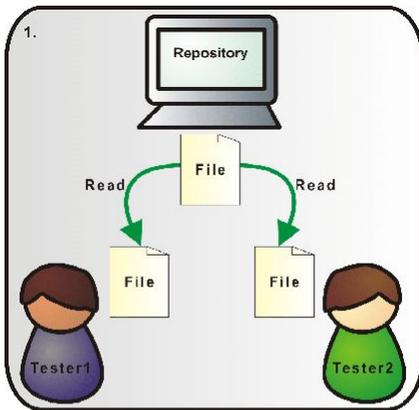
ВИДЫ **VCS**: ПО МЕХАНИЗМУ РАБОТЫ С ФАЙЛАМИ



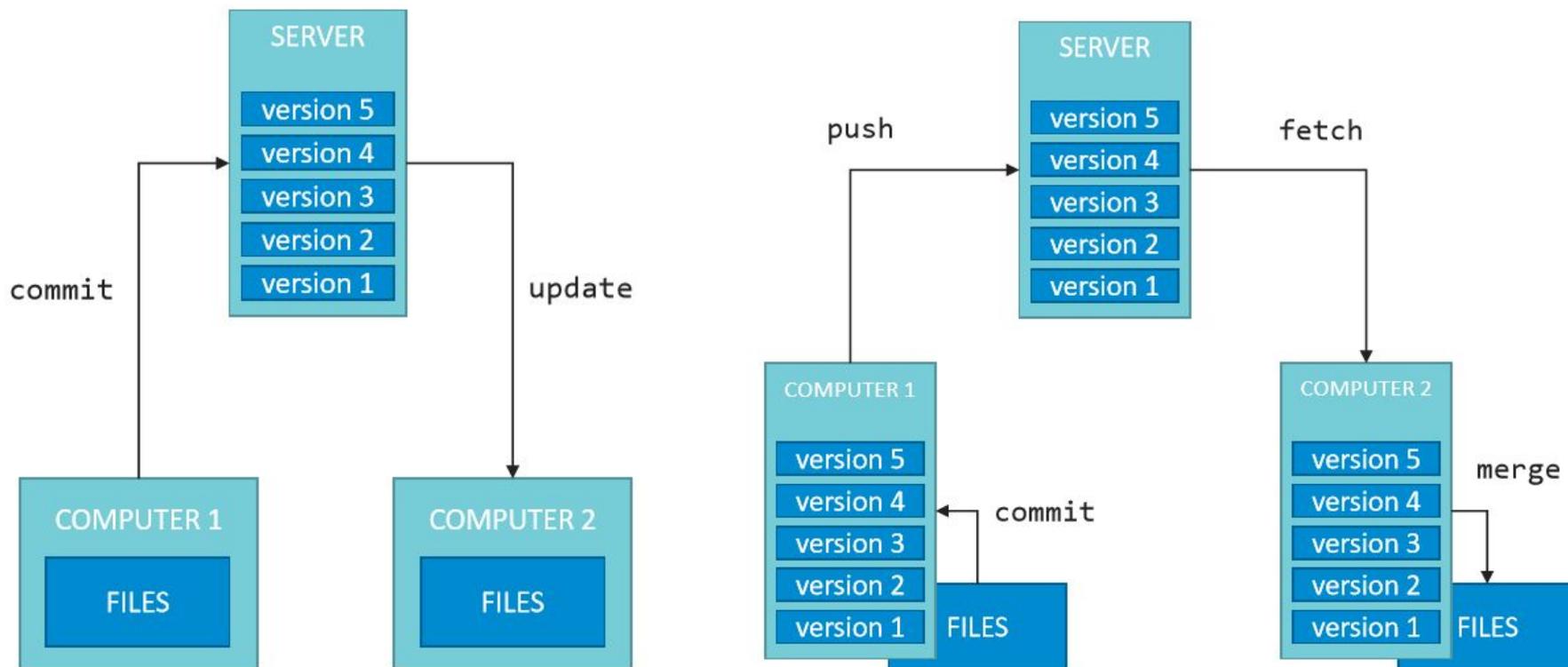
ВИДЫ VCS: LOCK-MODIFY-UNLOCK



ВИДЫ VCS: COPY-MODIFY-MERGE



ВИДЫ **VCS**: ЦЕНТРАЛИЗОВАННАЯ ИЛИ РАСПРЕДЕЛЕННАЯ

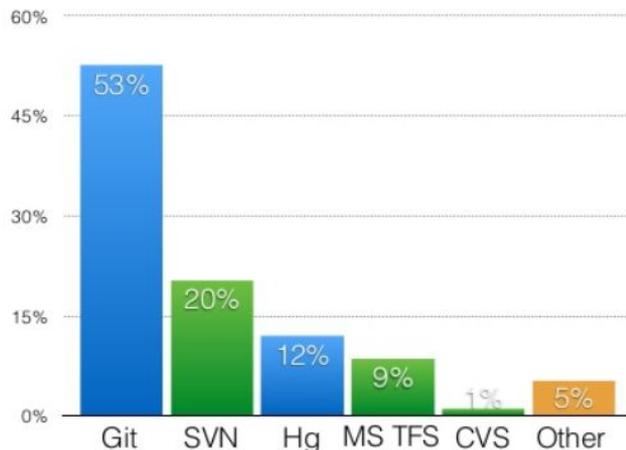


ПОЧЕМУ GIT



В 2005 году Линус Торвалдс разработал GIT для управления версиями ядра Linux, однако, чуть позднее его поддержкой занялся другой человек, японский инженер-программист – Джунио Хамано. На сегодняшний день, GIT – это одна из самых известных систем контроля версий с открытым исходным кодом, на которую полагаются миллионы проектов по всем миру (включая как коммерческие, так и бесплатные проекты). GIT – это полностью бесплатное программное обеспечение, поддерживающее множество ОС, таких как Mac, Linux, Windows и Solaris. Скачать соответствующие дистрибутивы можно [о сайта GIT.](#)

Git is the most used VCS



ПРЕИМУЩЕСТВА GIT

- ✓ Распределенная система управления версиями, GIT следует принципу одноранговой сети – **peer to peer** (равный к равному) в отличие от других систем вроде Subversion (SVN), которая основана на модели **client-server** (клиент-сервер).
- ✓ GIT позволяет разработчикам иметь множество совершенно независимых веток кода. Создание, удаление и объединение этих веток происходит без каких-либо проблем и больших затрат времени
- ✓ В GIT все операции атомарны; это означает, что любое действие может быть полностью удачным или провалиться (без каких-либо изменений). Это действительно важно, так как в некоторых системах контроля версий (вроде CVS), где действия не атомарны, некоторые повисшие операции по всему хранилищу, могут оставить его в нестабильном состоянии.
- ✓ В отличие от других VCS, таких как SVN или CVS где метаданные хранятся в скрытых папках (.cvs, .svn, и т.д.), в GIT все данные расположены в каталогах .git.
- ✓ Он использует модель данных, которая помогает обеспечить криптографическую целостность всего, что присутствует в репозитории. Каждый раз когда файлы добавляются или коммитятся, генерируются их контрольные суммы; аналогичный процесс происходит при их извлечении.
- ✓ Еще одна превосходная функция, присутствующая в GIT – это его **индекс**. В пределах индекса, разработчики могут форматировать коммиты и просматривать их до фактического применения.

УСТАНОВКА И НАСТРОЙКА GIT

1. Скачать установочный файл: <http://git-scm.com/downloads>
2. Выполнить все шаги, используя дефолтные опции

Command Prompt

```
Microsoft Windows [Version 10.0.17763.379]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\ihar.pryshchepau>git version
git version 2.16.1.windows.1
```

3. Открыть консоль git bash и назначить имя и емейл вашего пользователя

- ❑ \$ MINGW64 ~ git config --global user.name "Ivan Petrov"
- ❑ \$ MINGW64 ~ git config --global user.email ivan.petrov@gmail.com

4. Сгенерировать ssh ключ для работы с репозиторием по протоколу ssh

- ❑ \$ MINGW64 ~ ssh-keygen -t rsa -C "ivan.petrov@gmail.com"

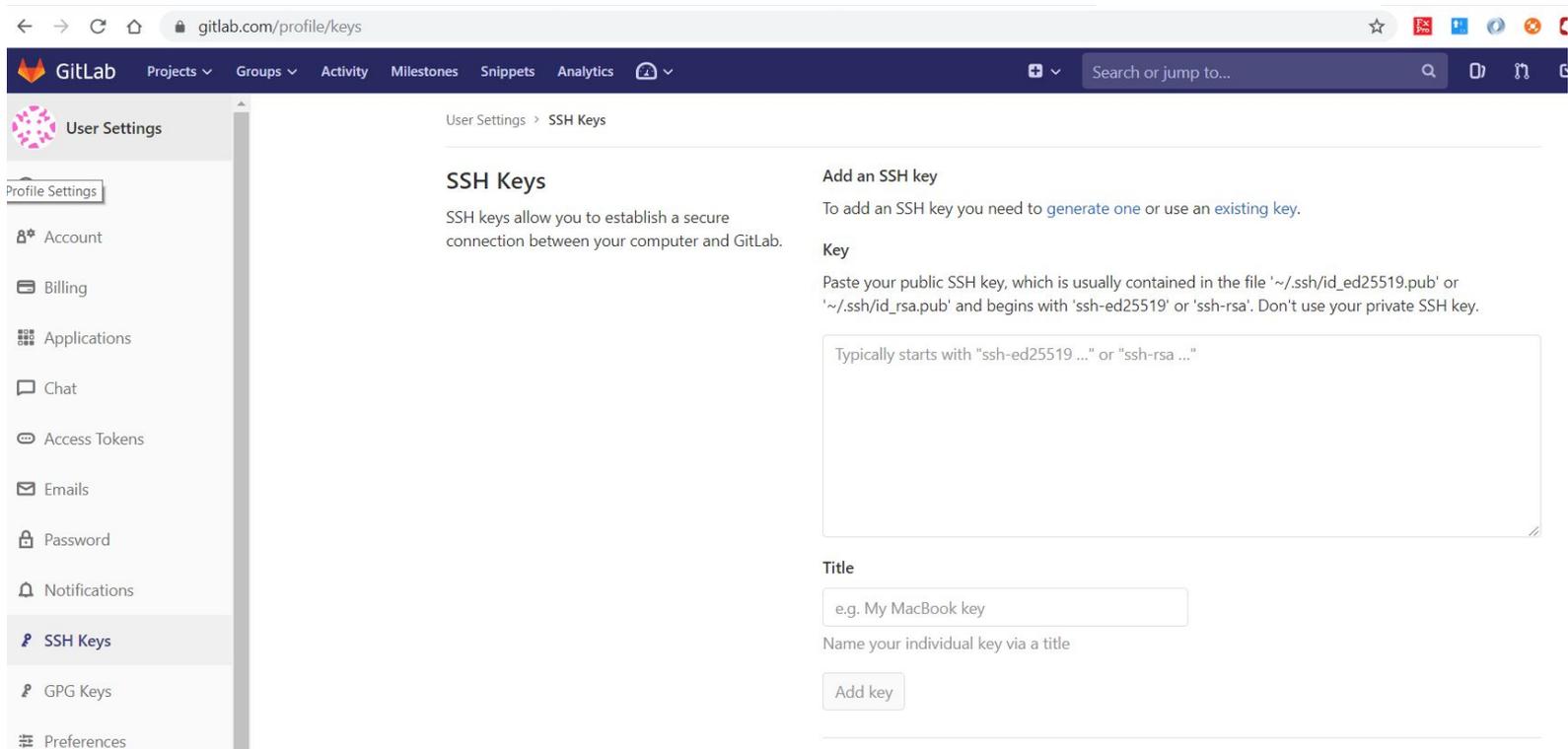
Name	Ext	Size	Date	Attr
[..]		<DIR>	08/14/2019 13:08	----
id_rsa		3,326	08/14/2019 13:08	-a--
id_rsa	pub	746	08/14/2019 13:08	-a--
known_hosts		187	08/14/2019 12:53	-a--



GITLAB REPOSITORY: CREATE & CLONE

15

1. Создать аккаунт в: <https://gitlab.com>
2. Добавить в профиль ssh public key



The screenshot shows the GitLab user settings page for SSH Keys. The browser address bar displays `gitlab.com/profile/keys`. The navigation bar includes links for Projects, Groups, Activity, Milestones, Snippets, and Analytics. The left sidebar shows the 'User Settings' menu with 'SSH Keys' selected. The main content area is titled 'SSH Keys' and includes a description: 'SSH keys allow you to establish a secure connection between your computer and GitLab.' Below this, there is a section for 'Add an SSH key' with instructions: 'To add an SSH key you need to generate one or use an existing key.' The 'Key' field is a large text area with a placeholder: 'Typically starts with "ssh-ed25519 ..." or "ssh-rsa ..."'. The 'Title' field is a text input with the example 'e.g. My MacBook key' and a label: 'Name your individual key via a title'. An 'Add key' button is located at the bottom of the form.

3. Создать новый проект(репозиторий)

Blank project	Create from template	Import project	CI/CD for external repo
---------------	----------------------	----------------	-------------------------

Project name

Project URL

Project slug

Want to house several dependent projects under the same namespace? [Create a group.](#)

Project description (optional)

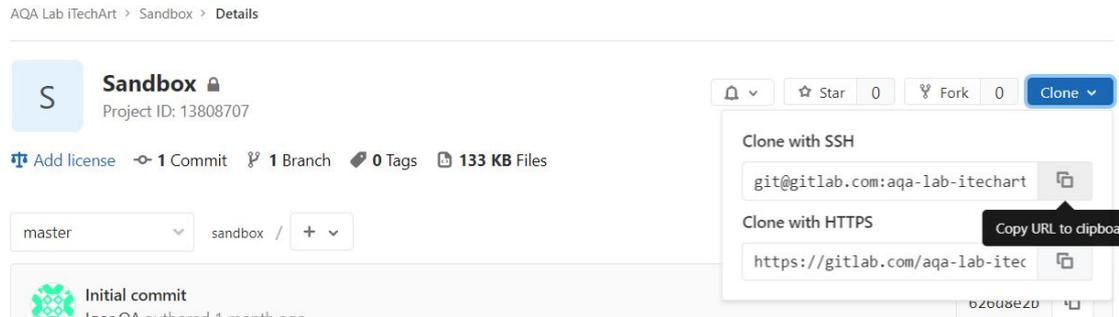
Visibility Level ⓘ

Private
Project access must be granted explicitly to each user.

Public
The project can be accessed without any authentication.

Initialize repository with a README
Allows you to immediately clone this project's repository. Skip this if you plan to push up an existing repository.

4. Скопировать в буфер обмена адрес репозитория для ssh



```
ihar.pryshchepau@nb-033-31 MINGW64 ~
$ cd '/c/Users/ihar.pryshchepau/Documents/EDU/AQA Lab/code/repos/Sandbox 4'

ihar.pryshchepau@nb-033-31 MINGW64 ~/Documents/EDU/AQA Lab/code/repos/Sandbox 4
$ git clone git@gitlab.com:aq-a-lab-itechart/sandbox.git
Cloning into 'sandbox'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 0), reused 0 (delta 0)
Receiving objects: 100% (3/3), done.

ihar.pryshchepau@nb-033-31 MINGW64 ~/Documents/EDU/AQA Lab/code/repos/Sandbox 4
$ ls
sandbox/

ihar.pryshchepau@nb-033-31 MINGW64 ~/Documents/EDU/AQA Lab/code/repos/Sandbox 4
$ cd sandbox

ihar.pryshchepau@nb-033-31 MINGW64 ~/Documents/EDU/AQA Lab/code/repos/Sandbox 4/sandbox (master)
$ ls
Employees.csv

ihar.pryshchepau@nb-033-31 MINGW64 ~/Documents/EDU/AQA Lab/code/repos/Sandbox 4/sandbox (master)
$ |
```

GIT: COMMIT AND PUSH

- `$ git status` – показывает текущее состояние локального репозитория
- `$ git add <file name>` – добавляет изменения в git index
- `$ git commit -m <message>` – создает commit с комментарием в локальном репозитории
- `$ git push` – отправляет все существующие коммиты в локальном репозитории в удаленный репозиторий в соответствующую ветку

```
ihar.pryshchepau@nb-033-31 MINGW64 ~/Documents/EDU/AQA Lab/repos/poem (master)
$ git status
On branch master
Your branch is up to date with 'origin/master'.

Untracked files:
  (use "git add <file>..." to include in what will be committed)

        tobeornottobe.txt

nothing added to commit but untracked files present (use "git add" to track)

ihar.pryshchepau@nb-033-31 MINGW64 ~/Documents/EDU/AQA Lab/repos/poem (master)
$ git add tobeornottobe.txt

ihar.pryshchepau@nb-033-31 MINGW64 ~/Documents/EDU/AQA Lab/repos/poem (master)
$ git status
On branch master
Your branch is up to date with 'origin/master'.

Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

        new file:   tobeornottobe.txt

ihar.pryshchepau@nb-033-31 MINGW64 ~/Documents/EDU/AQA Lab/repos/poem (master)
$ git commit -m "Add file tobeornottobe.txt with 2 lines"
[master 4fcefca] Add file tobeornottobe.txt with 2 lines
1 file changed, 2 insertions(+)
create mode 100644 tobeornottobe.txt

ihar.pryshchepau@nb-033-31 MINGW64 ~/Documents/EDU/AQA Lab/repos/poem (master)
$ git push
Counting objects: 3, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 377 bytes | 377.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0)
To gitlab.com:iproff2011/poem.git
   0f7a9a5..4fcefca  master -> master

ihar.pryshchepau@nb-033-31 MINGW64 ~/Documents/EDU/AQA Lab/repos/poem (master)
$ |
```

GIT: FETCH FROM REMOTE AND MERGE

- `$ git fetch` – собирает все коммиты из целевой ветки, которых нет в текущей ветке, и сохраняет их в локальном репозитории, не сливая их в текущую ветку.
- `$ git merge` – добавляет изменения в текущую ветку

```
ihar.pryshchepau@nb-033-31 MINGW64 ~/Documents/EDU/AQA Lab/repos/poem (master)
$ git status
On branch master
Your branch is up to date with 'origin/master'.

nothing to commit, working tree clean

ihar.pryshchepau@nb-033-31 MINGW64 ~/Documents/EDU/AQA Lab/repos/poem (master)
$ git fetch
remote: Enumerating objects: 5, done.
remote: Counting objects: 100% (5/5), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 0), reused 0 (delta 0)
Unpacking objects: 100% (3/3), done.
From gitlab.com:iproff2011/poem
 17129fe..851a12b master    -> origin/master

ihar.pryshchepau@nb-033-31 MINGW64 ~/Documents/EDU/AQA Lab/repos/poem (master)
$ git status
On branch master
Your branch is behind 'origin/master' by 1 commit, and can be fast-forwarded.
  (use "git pull" to update your local branch)

nothing to commit, working tree clean

ihar.pryshchepau@nb-033-31 MINGW64 ~/Documents/EDU/AQA Lab/repos/poem (master)
$ git merge
Updating 17129fe..851a12b
Fast-forward
 README.md | 2 ++
 1 file changed, 2 insertions(+)

ihar.pryshchepau@nb-033-31 MINGW64 ~/Documents/EDU/AQA Lab/repos/poem (master)
$ git status
On branch master
Your branch is up to date with 'origin/master'.

nothing to commit, working tree clean
```

GIT: PULL FROM REMOTE

- \$ git pull – добавляет все коммиты в текущую ветку без возможности их просмотра

```
ihar.pryshchepau@nb-033-31 MINGW64 ~/Documents/EDU/AQA Lab/repos/poem (master)
$ git status
On branch master
Your branch is up to date with 'origin/master'.

nothing to commit, working tree clean

ihar.pryshchepau@nb-033-31 MINGW64 ~/Documents/EDU/AQA Lab/repos/poem (master)
$ git pull
remote: Enumerating objects: 5, done.
remote: Counting objects: 100% (5/5), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 0), reused 0 (delta 0)
Unpacking objects: 100% (3/3), done.
From gitlab.com:iproff2011/poem
   851a12b..5db083c master    -> origin/master
Updating 851a12b..5db083c
Fast-forward
 README.md | 6 +-----
 1 file changed, 1 insertion(+), 5 deletions(-)

ihar.pryshchepau@nb-033-31 MINGW64 ~/Documents/EDU/AQA Lab/repos/poem (master)
$ git status
On branch master
Your branch is up to date with 'origin/master'.

nothing to commit, working tree clean
```

GIT GUI & GITK

Git Gui (poem) C:/Users/ihtar.pryshchepau/Documents/EDU/AQA Lab/repos/poem

Repository Edit Branch Commit Merge Remote Tools Help

Current Branch: master

Unstaged Changes Modified, not staged File: tobeornottobe.txt

```
@@ -2,5 +2,7 @@ To be, or not to be, that is the question:
Whether 'tis nobler in the mind to suffer
The slings and arrows of outrageous fortune,
Or to take Arms against a Sea of troubles,
And by opposing end them: to die, to sleep
No more; and by a sleep, to say we end
+The heart-ache, and the thousand natural shocks
+That Flesh is heir to? 'Tis a consummation
```

Staged Changes (Will Commit)

Commit Message: New Commit Amend Last Commit

Rescan
Stage Changed
Sign Off
Commit
Push

Ready.

poem: All files - gitk

File Edit View Help

Local uncommitted changes: not checked in to index

- ← master (remotes/origin/master) Update README.md
- ↑ Update README.md
- ◆ Add 2 more lines
- ◆ Add 2 more lines
- ◆ Update README.md
- ◆ Add file tobeornottobe.txt with 2 lines
- ◆ Remove file
- ◆ Add 2 lines of poem
- ◆ Initial commit

SHA1 ID	Author	Date
951a12b0b422aa92185b13ec5260f0e6636d0ad9	Ihar Pryshchepau <iproff2011@gmail.com>	2019-09-24 16:30:15
17429f6c1646a18f681f25817451d5077bcefe	Ihar Pryshchepau <iproff2011@gmail.com>	2019-09-24 16:07:41
5cb913c5580c447d9f15463a7d0a057eb3f958aa	Ihar Pryshchepau <iproff2011@gmail.com>	2019-09-24 13:00:54
7429f6c1646a18f681f25817451d5077bcefe	Ihar Pryshchepau <iproff2011@gmail.com>	2019-09-24 15:58:17
17429f6c1646a18f681f25817451d5077bcefe	Ihar Pryshchepau <iproff2011@gmail.com>	2019-09-24 13:07:53
5cb913c5580c447d9f15463a7d0a057eb3f958aa	Ihar Pryshchepau <iproff2011@gmail.com>	2019-09-24 12:30:54
7429f6c1646a18f681f25817451d5077bcefe	Ihar Pryshchepau <iproff2011@gmail.com>	2019-09-24 12:30:58
17429f6c1646a18f681f25817451d5077bcefe	Ihar Pryshchepau <iproff2011@gmail.com>	2019-09-24 12:25:36
5cb913c5580c447d9f15463a7d0a057eb3f958aa	Ihar Pryshchepau <iproff2011@gmail.com>	2019-09-24 12:22:59
17429f6c1646a18f681f25817451d5077bcefe	Ihar Pryshchepau <iproff2011@gmail.com>	2019-09-24 12:11:16

SHA1 ID: 951a12b0b422aa92185b13ec5260f0e6636d0ad9 Row: 3 / 10

Find: commit [containing:]

Search Patch Tree

Diff Old version New version Lines of context: 3 Ignore space changes

Author: Ihar Pryshchepau <iproff2011@gmail.com> 2019-09-24 16:07:41
Committer: Ihar Pryshchepau <iproff2011@gmail.com> 2019-09-24 16:07:41
Parent: 17429f6c1646a18f681f25817451d5077bcefe (Add 2 more lines)
Child: 5cb913c5580c447d9f15463a7d0a057eb3f958aa (Update README.md)
Branches: master, remotes/origin/master
Follows:
Precedes:

Update README.md

----- README.md

```
index 6358e02..e1c132e 100644
@@ -1,3 +1,5 @@
 # Poem
+Hi there
+
```

Working directory

```
git checkout -- file.txt  
git checkout .  
git clean -xdf
```

Staging area (Index)

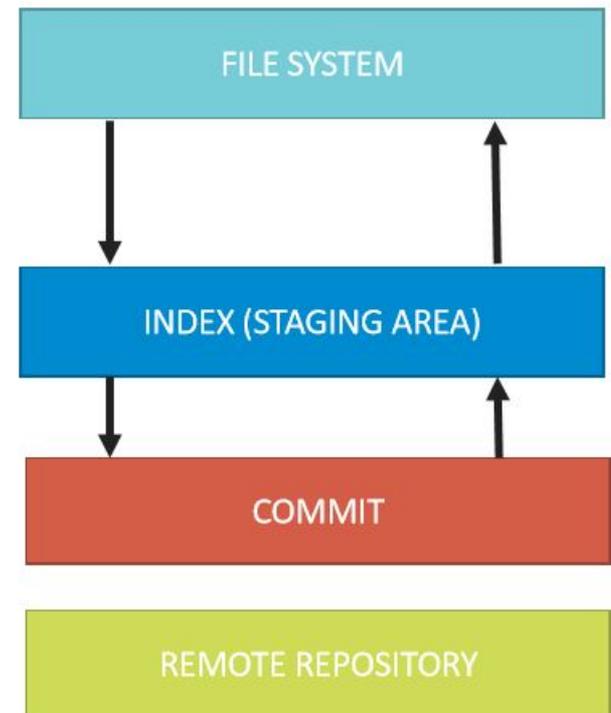
```
git reset -- file.txt
```

Local branch

```
git reset HEAD^^ (HEAD~2)  
git commit --amend -m "commit message"
```

Remote repository

```
git revert <sha1>
```



.gitignore

```
# no .log files
*.log

# but do track error.log, even though you're
ignoring .log files above
!error.log

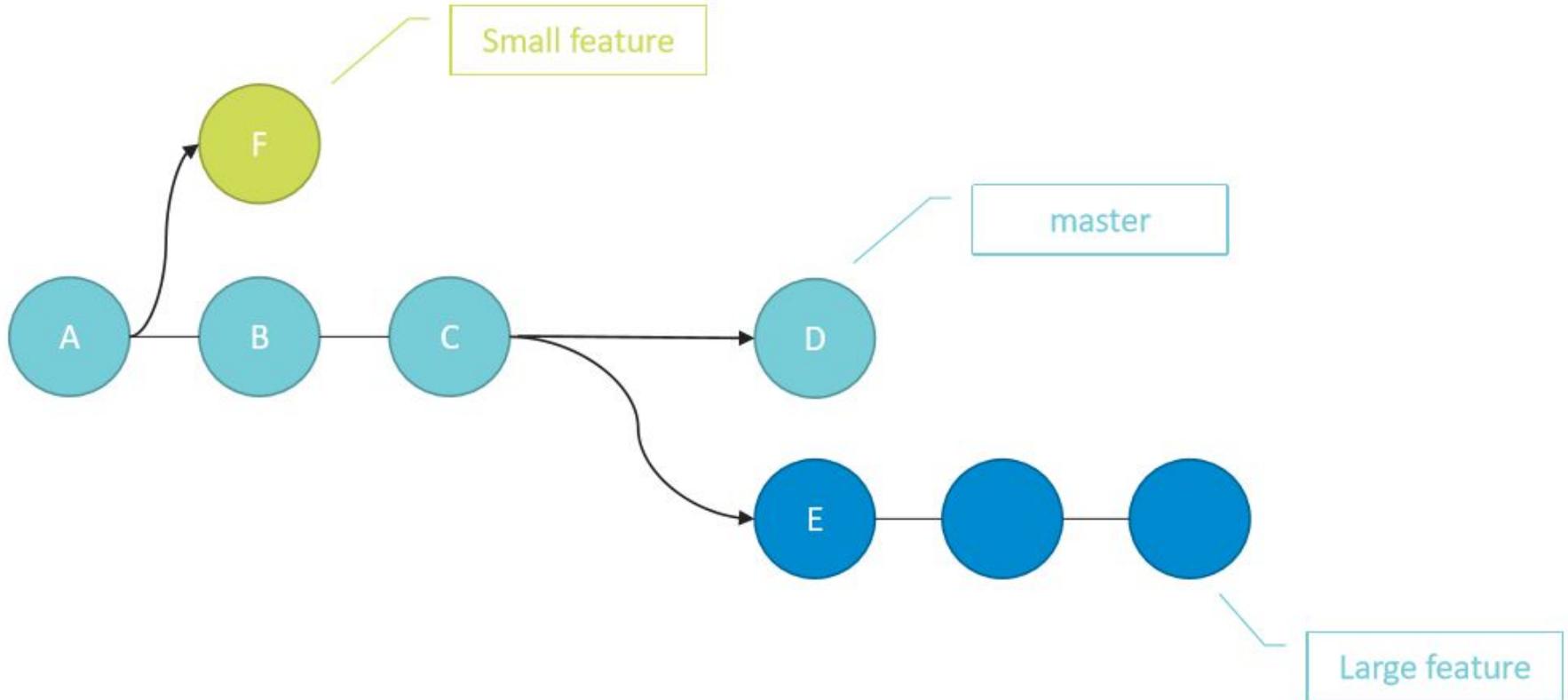
# only ignore the TODO file in the current
directory, not subdir/TODO
/TODO

# ignore all files in the build/ directory
build/

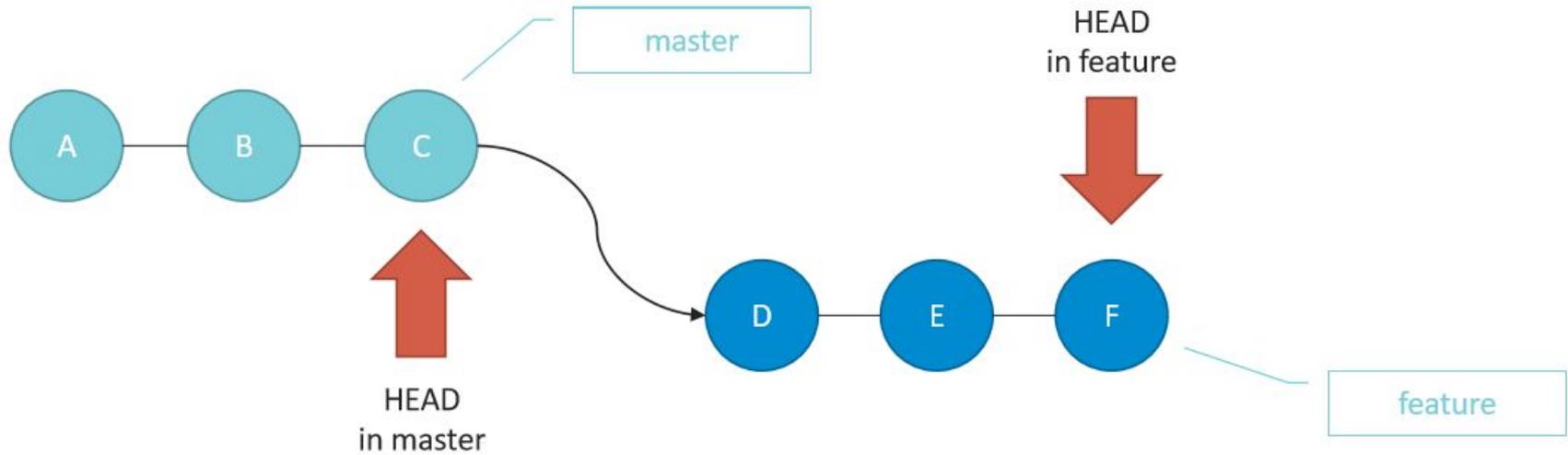
# ignore doc/notes.txt, but not doc/server/arch.txt
doc/*.txt

# ignore all .pdf files in the doc/ directory
doc/**/*.pdf
```

GIT: BRANCH CONCEPT

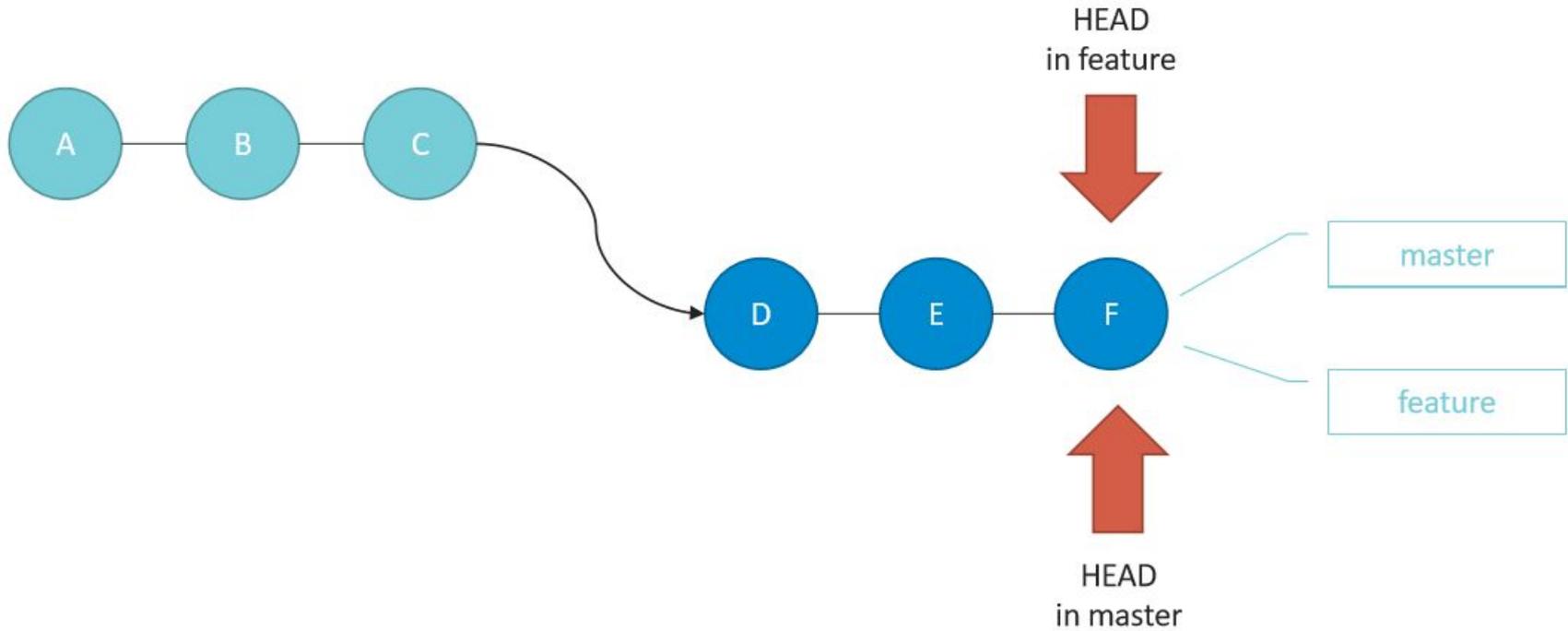


Fast-forward merge

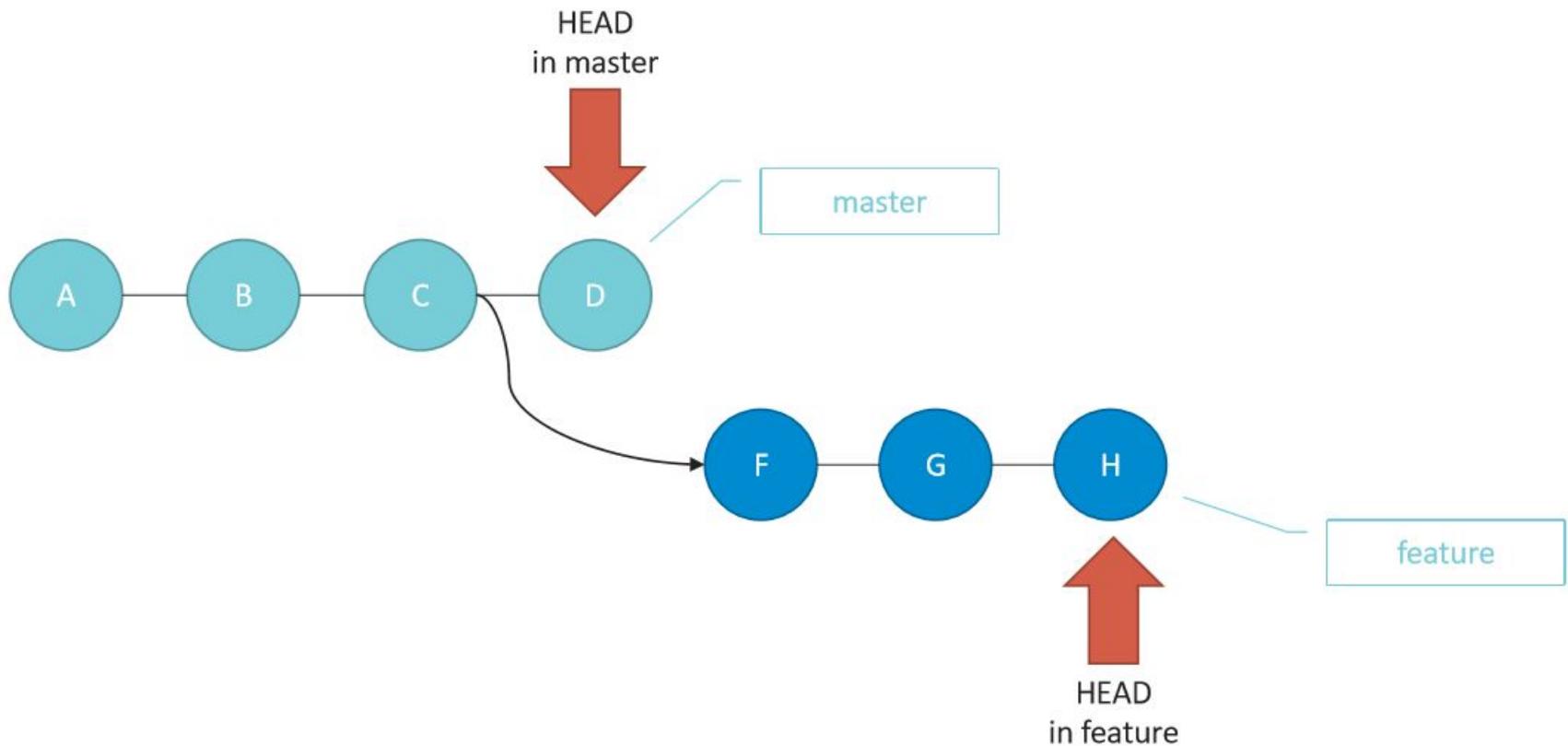


GIT: BRANCH MERGE

Fast-forward merge

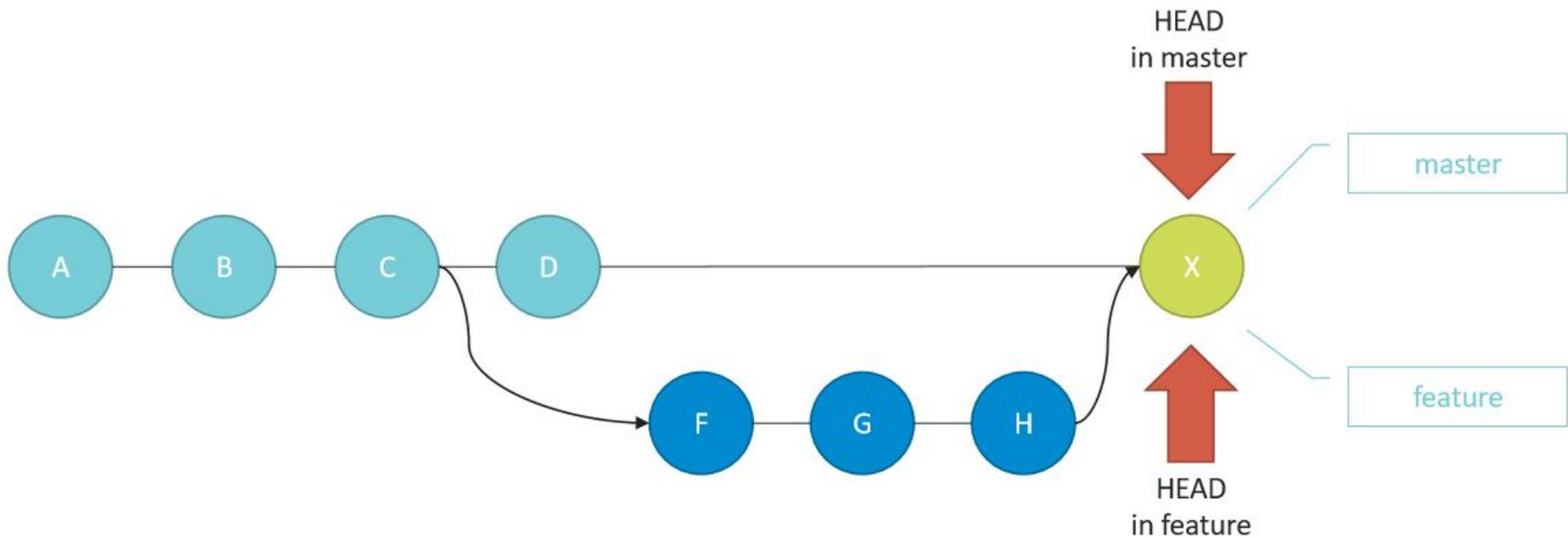


Non fast-forward merge

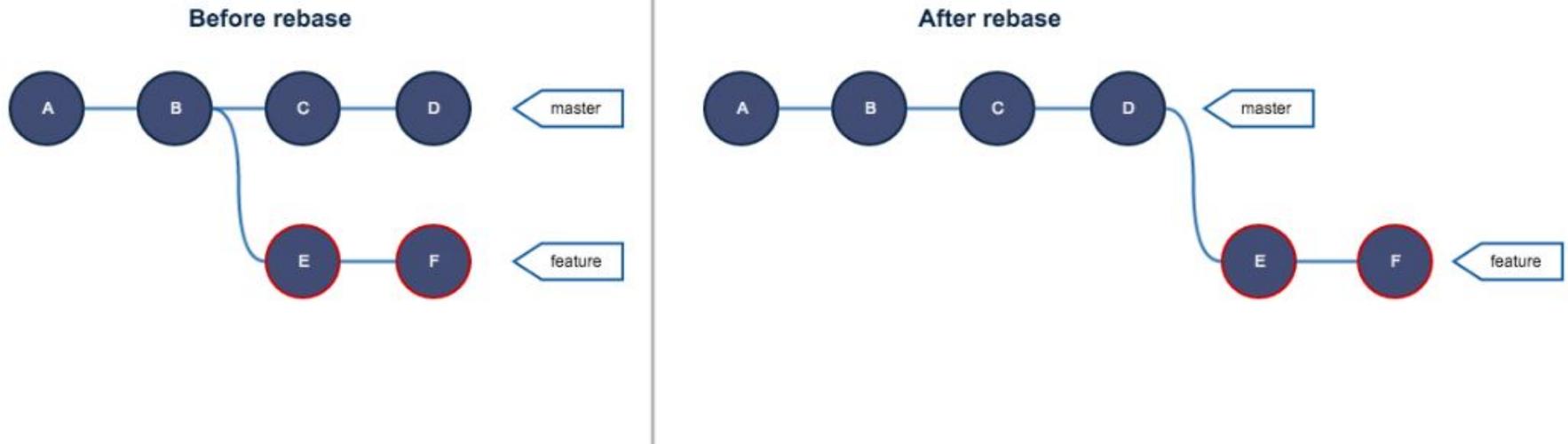


GIT: BRANCH MERGE

Non fast-forward merge



GIT: BRANCH REBASE



**ЗОЛОТОЕ ПРАВИЛО:
!!!НИКОГДА НЕ ДЕЛАЙТЕ REBASE ОБЩИХ ВЕТОК**

SOLVE CONFLICT

Abort merge

```
git merge --abort
```

Resolve by selecting version

```
git checkout --ours <branch name> <file name>
```

Resolve manually

```
git diff
```

Undo merge

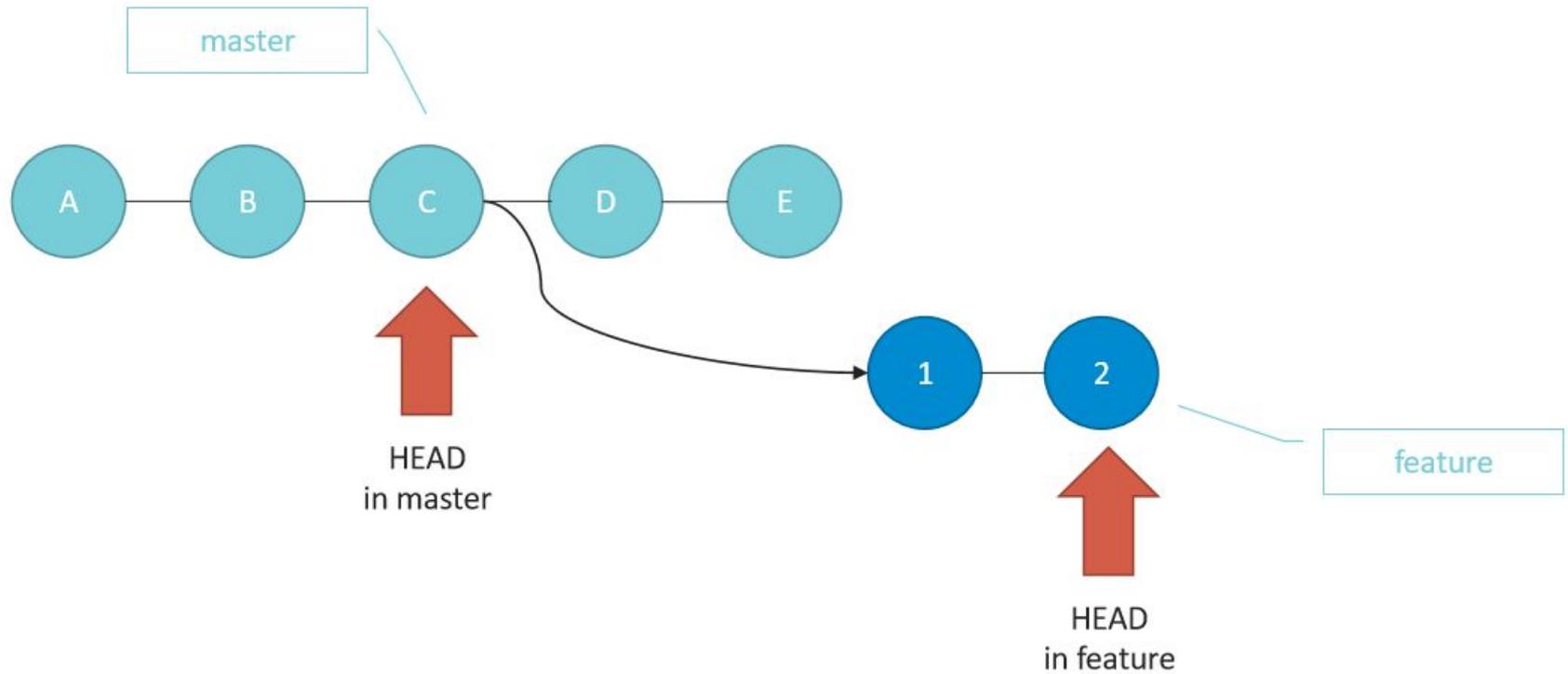
```
git revert 09fe472
```

User merge tool

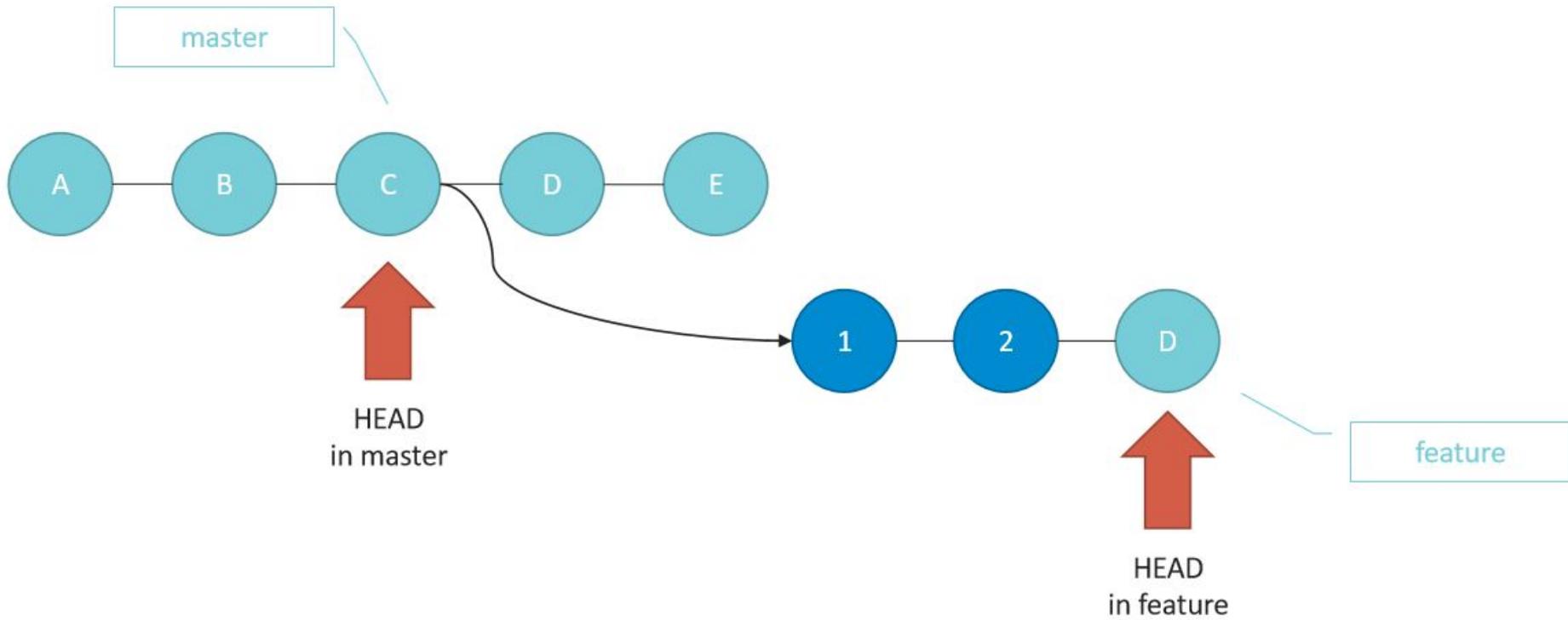
AVOID CONFLICT

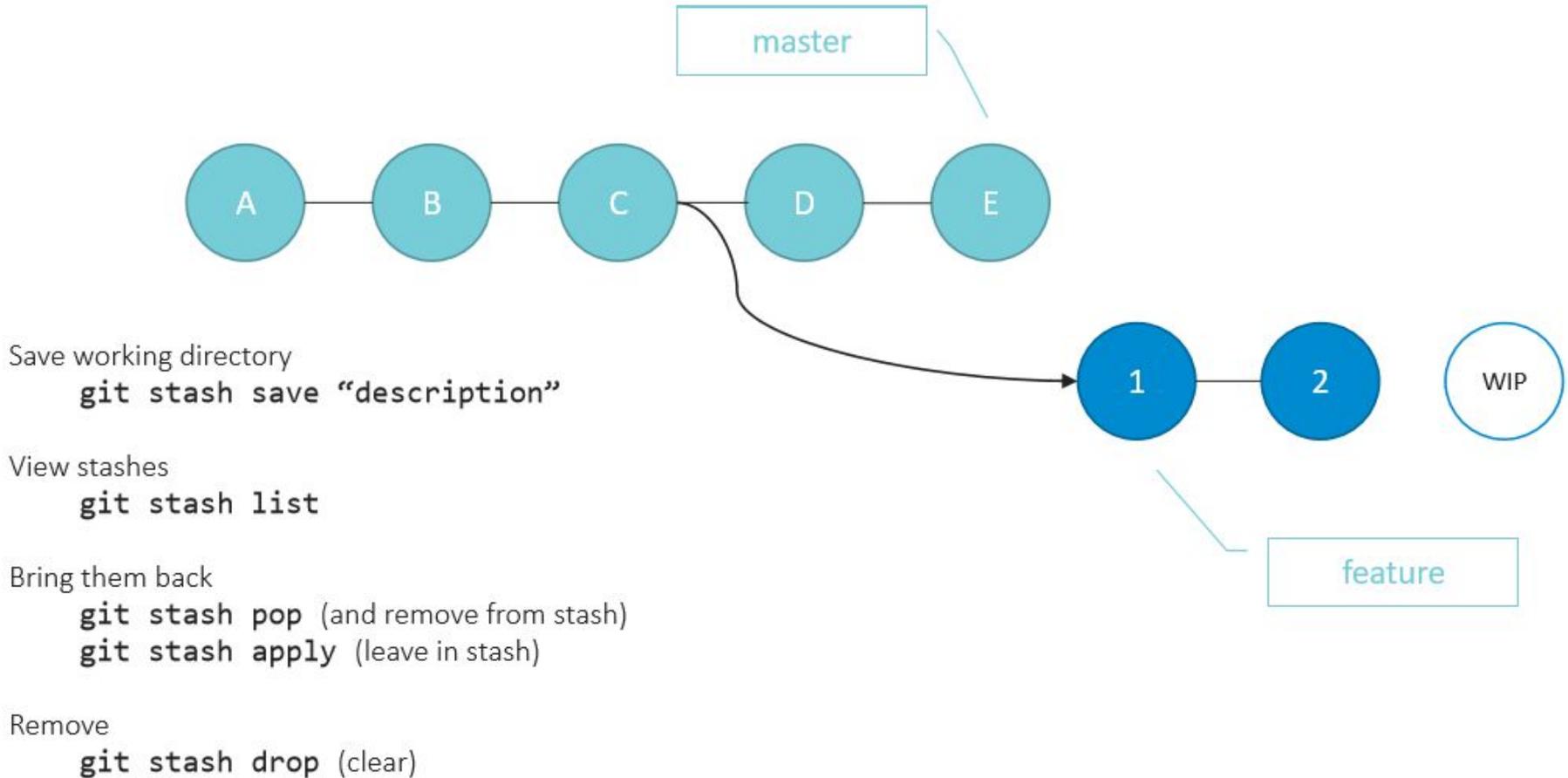
- Short commits
- No edits to whitespaces
- Merge often

GIT: CHERRY PICK



GIT: CHERRY PICK





GIT: ADD LOCAL PROJECT TO REMOTE

34

- \$ git init – инициализирует текущую папку как локальный репозиторий git

```
MINGW64:/c:/Users/ihar.pryshchepau/Documents/EDU/AQA Lab/code/repos/...
ihar.pryshchepau@nb-033-31 MINGW64 ~/Documents/EDU/AQA Lab/code/repos/GitLab2
$ git init
Initialized empty Git repository in C:/Users/ihar.pryshchepau/Documents/EDU/AQA
Lab/code/repos/GitLab2/.git/

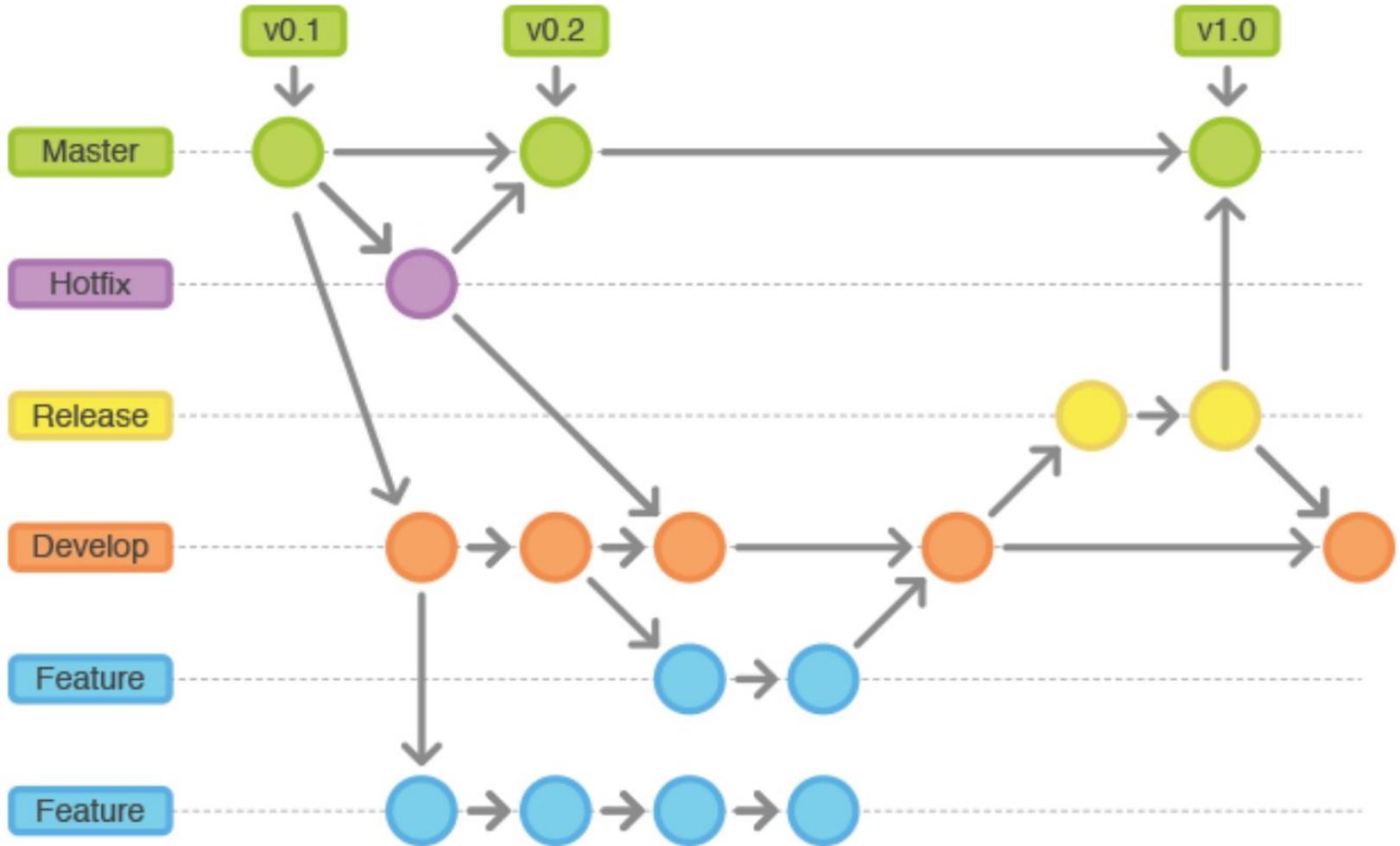
ihar.pryshchepau@nb-033-31 MINGW64 ~/Documents/EDU/AQA Lab/code/repos/GitLab2 (m
aster)
$ git add .

ihar.pryshchepau@nb-033-31 MINGW64 ~/Documents/EDU/AQA Lab/code/repos/GitLab2 (m
aster)
$ git commit -m "Add files to the new repository"
[master (root-commit) 6a0489d] Add files to the new repository
1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 my_project.txt

ihar.pryshchepau@nb-033-31 MINGW64 ~/Documents/EDU/AQA Lab/code/repos/GitLab2 (m
aster)
$ git push --set-upstream git@gitlab.com:iproff2011/example.git master
Counting objects: 3, done.
Writing objects: 100% (3/3), 234 bytes | 234.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0)
To gitlab.com:iproff2011/example.git
 * [new branch]      master -> master
Branch 'master' set up to track remote branch 'master' from 'git@gitlab.com:iprof
f2011/example.git'.

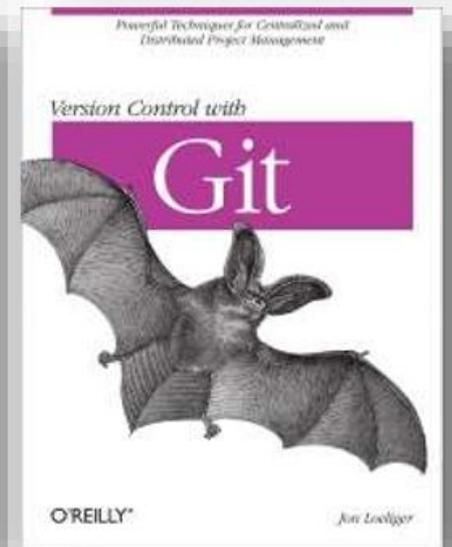
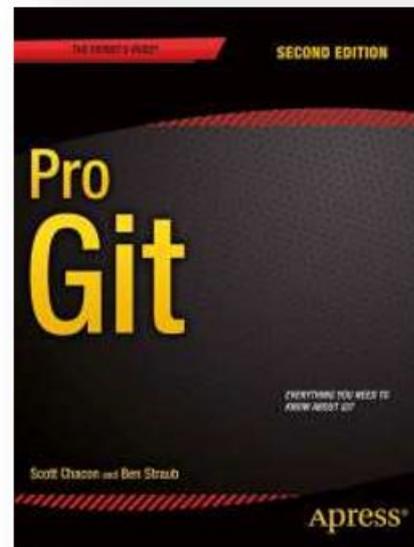
ihar.pryshchepau@nb-033-31 MINGW64 ~/Documents/EDU/AQA Lab/code/repos/GitLab2 (ma
ster)
$ |
```

GIT FLOW



- `git config --global user.name "<user name>"`
- `git config --global user.email "vitali_shulha@epam.com"`
- `git config --global core.editor "'C:/Program Files (x86)/Notepad++/notepad++.exe'"`
- `git blame`
- `git bisect`
- `git log --pretty=oneline`
- `git log --pretty=format:"%h %s" --graph`
- `git config --global alias.last 'log -1 HEAD'`
- `git last`
- `git log master..experiment`
- `git filter-branch --tree-filter 'rm -f passwords.txt' HEAD`
- `git rerere`
- `git submodule`

- **Pro Git** by Scott Chacon and Ben Straub
- **Version Control with Git** by Jon Loeliger, Matthew McCullough



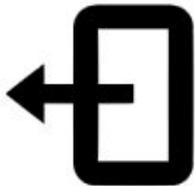
In case of fire



1. `git commit`



2. `git push`



3. leave building