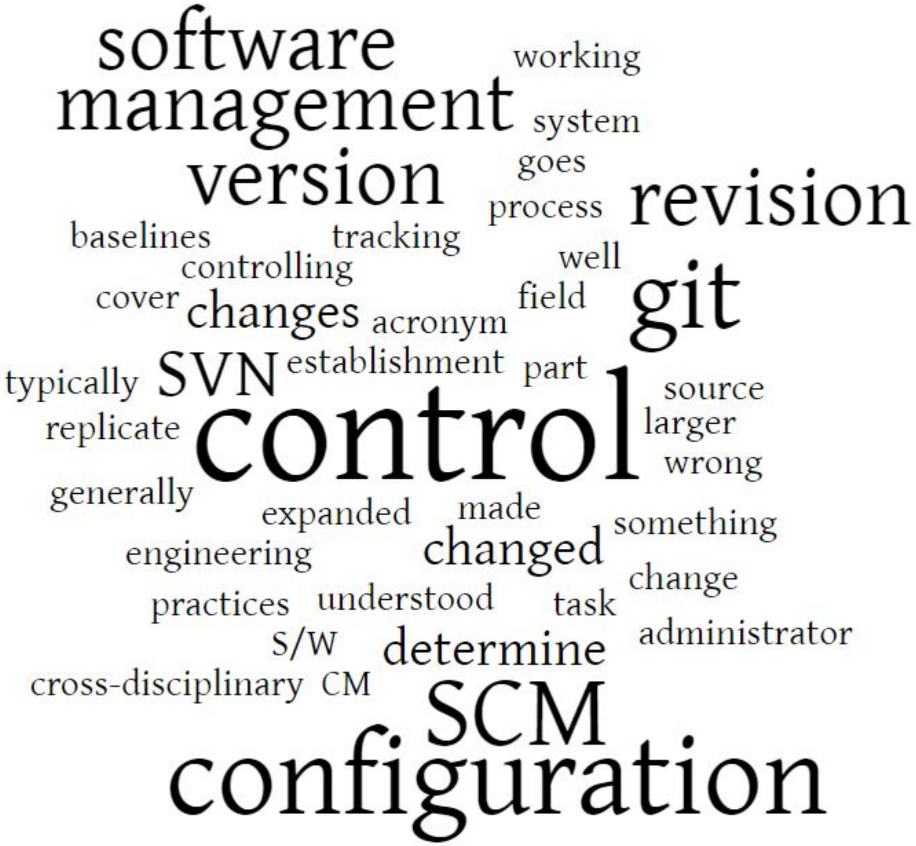
git intro

by Vyacheslav Koldovskyy

softserve

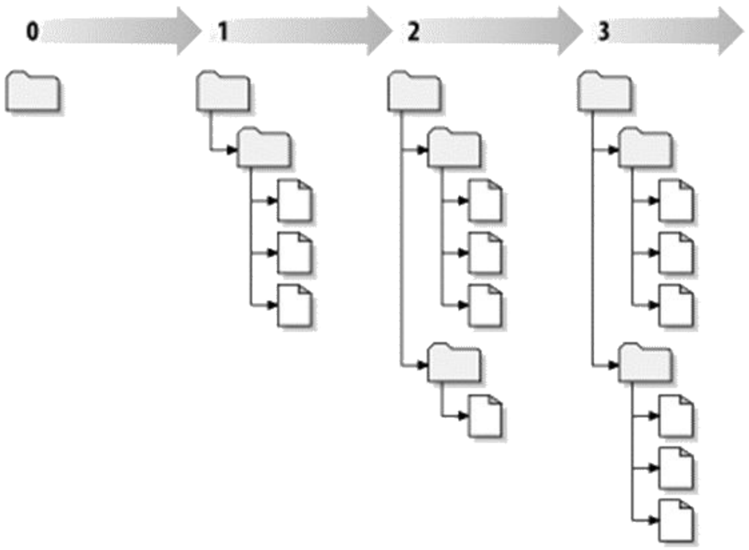
SCM/VC/RC/SC

A component **software configuration management (SCM), version control (VC)**, also known as **revision control (RC)** or **source control (SC)** is the management of changes to documents, computer programs, large web sites, and other collections of information.



Typical tasks for version control systems

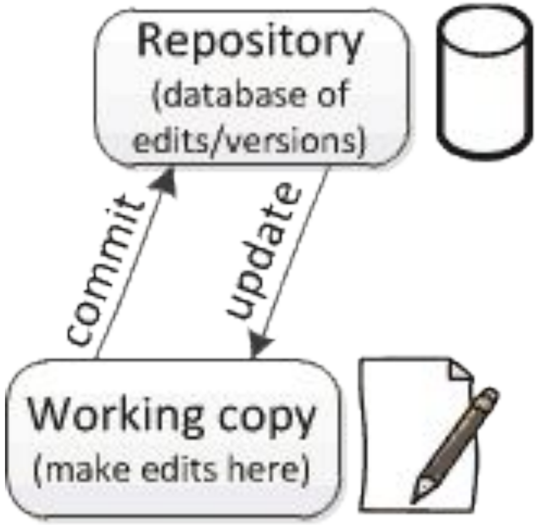
- Tracking changes
- Making updates
- Getting updates
- Resolving Conflicts
- Diffing (viewing differences)
- Branching and merging
- Controlling change sets

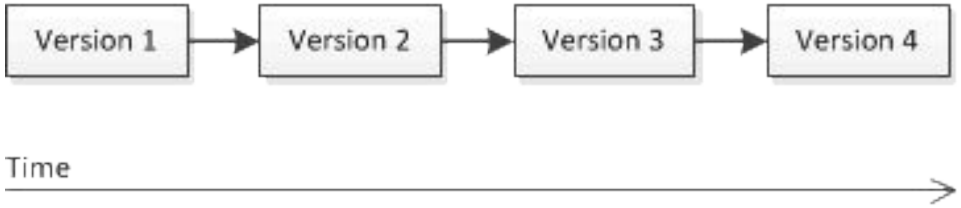


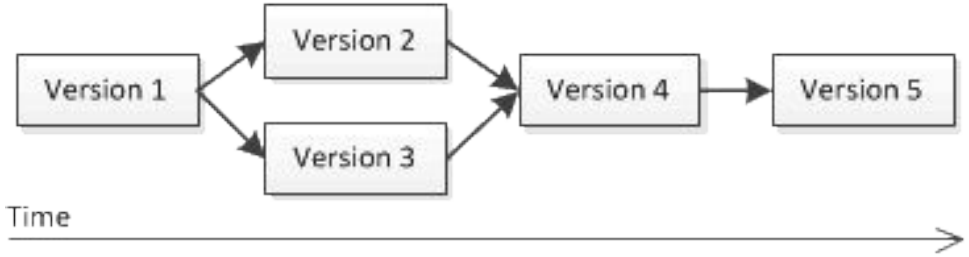
softserve

Terms

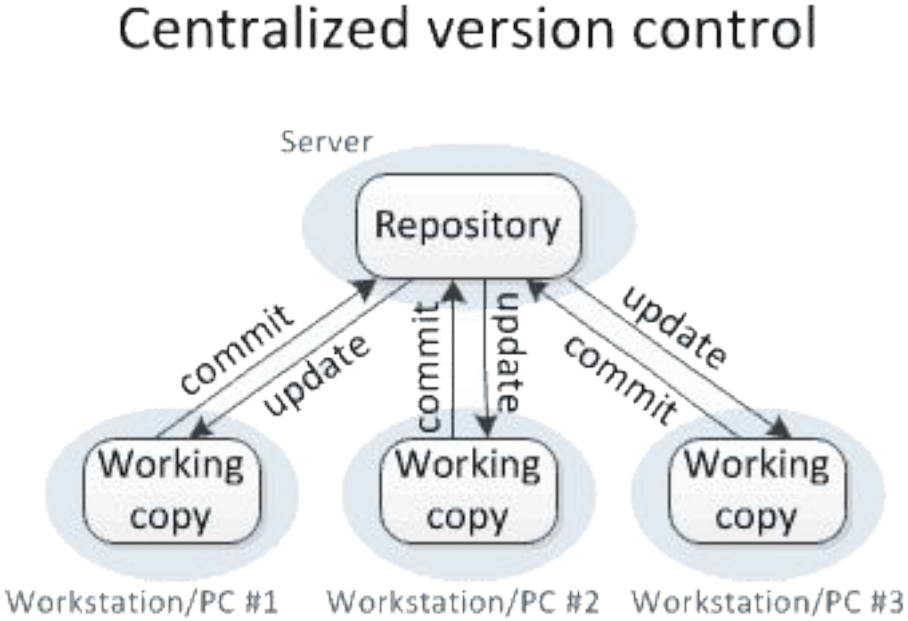
- Repository
- Working Copy
- Merging
- Version



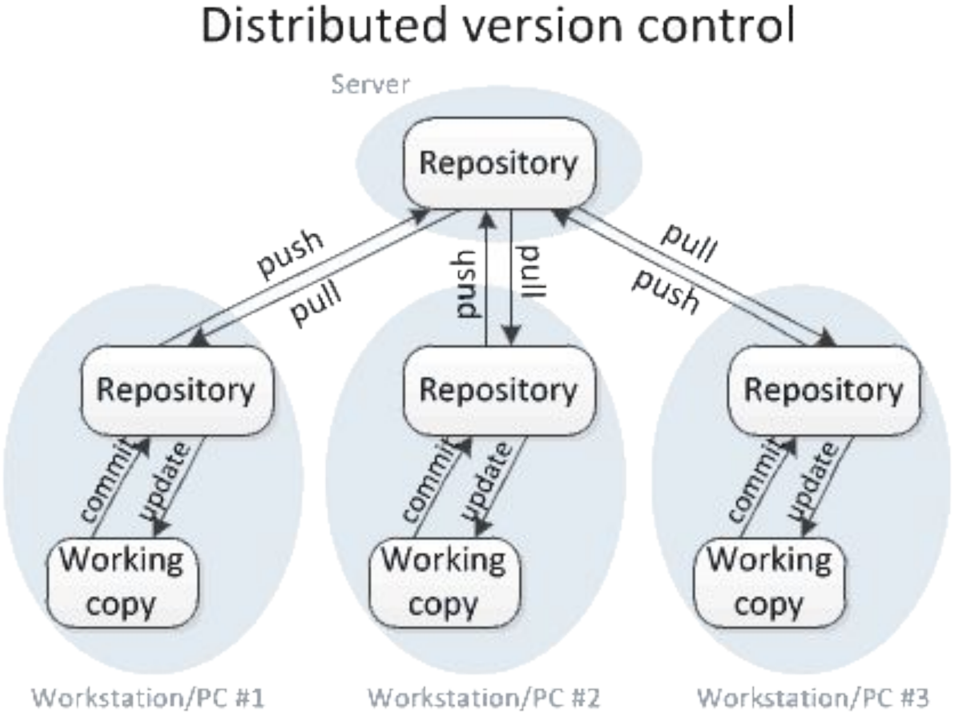




Types of Version Control Systems



CVS, Perforce, SVN,
Team Foundation Server (TFS)



git, mercurial

softserve

Git Intro

git – is a distributed version control system with an emphasis on speed, data integrity, and support for distributed, non-linear workflows.

git was initially designed and developed by *Linus Torvalds* for Linux kernel development in 2005, and has since become the most widely adopted version control system for software development.

Install git

Official website:
https://git-scm.com

Linux OS

Debian Family (Debian, Ubuntu, Mint)

```
#apt-get install git
```

Red Hat Family (RHEL, CentOS, Fedora)

```
#yum install git
```

MS Windows

https://git-scm.com/download/win

Mac OS

Step 1 - Install Homebrew

```
#ruby -e "$(curl -fsSL https://raw.githubusercontent.com/
Homebrew/install/master/install)"
brew doctor
```

Step 2 - Install git

```
#brew install git
```

Configure before use

Git comes with tool called **`git config`**

Identity

```
$ git config --global user.name “Jon Snow‟
$ git config --global user.email jon@example.com
```

Editor

```
$ git config --global core.editor emacs
```

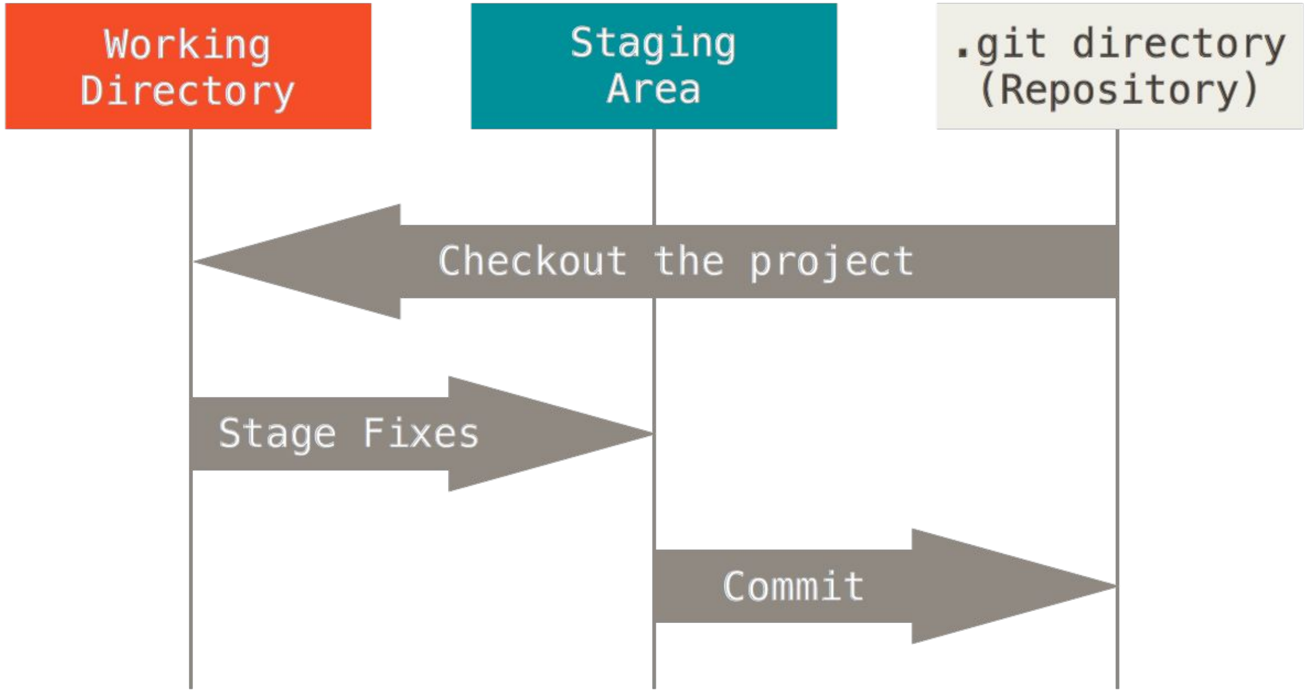
Check settings

```
$ git config --list
```

softserve

Basic terms

- Local repository stored in hidden folder .git
- Working directory - folder with code
- Commit - snapshot of working directory
- Staging area or Index -

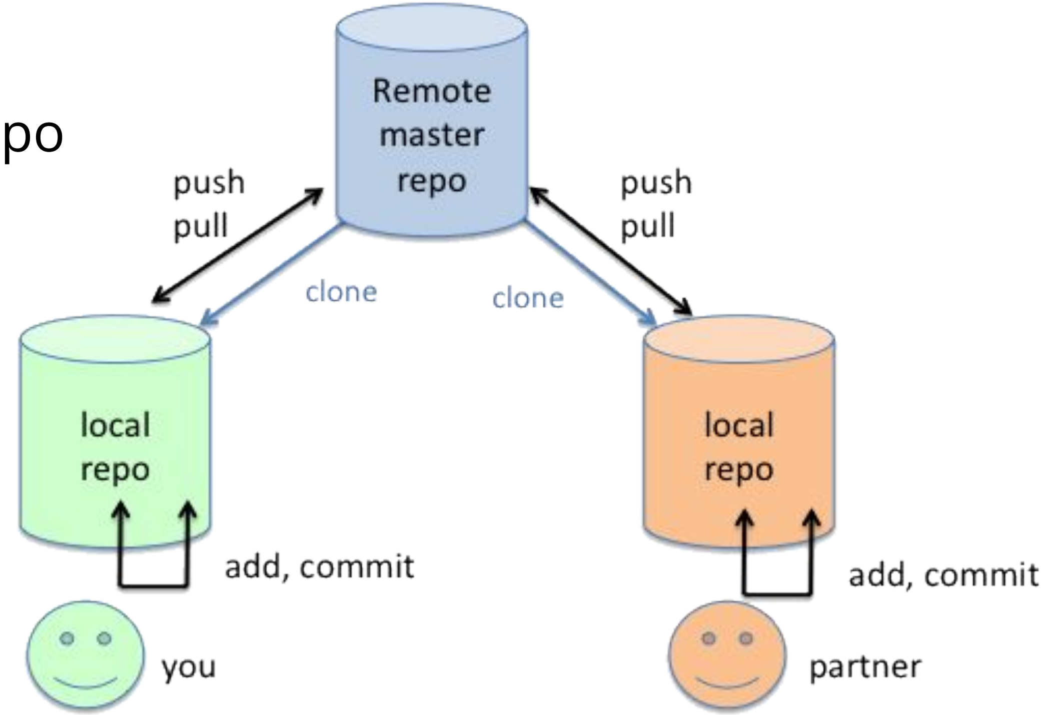


softserve

Create/clone repository

`git init` – create an empty local repo

`git clone <URL>` – create local repo from remote repo



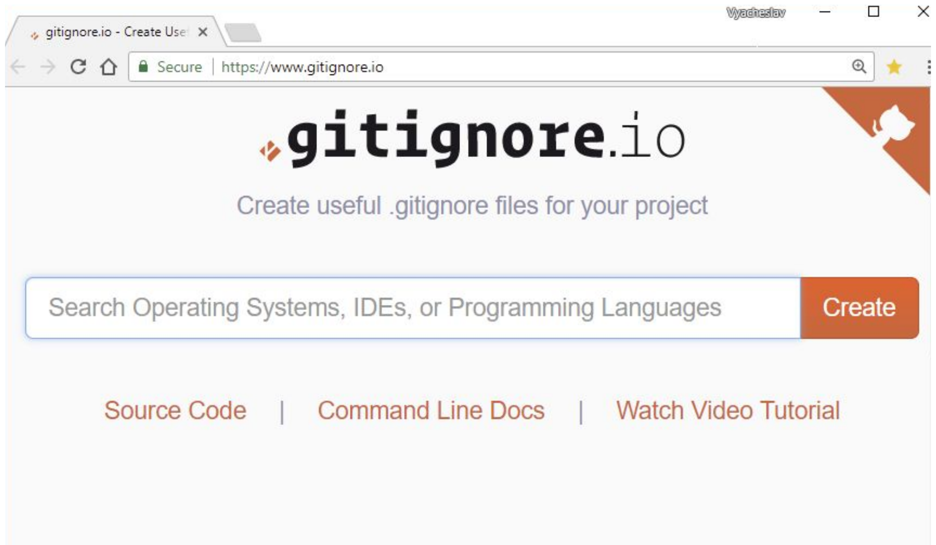
softserve

.gitignore

`.gitignore` - contains list of files and folders that are ignored by git in working folder

Typically ignored files:

- Operating system files (Thumbs.db, .DS_Store)
- Application/IDE configuration files (.vscode)
- Generated files (*.exe, *.min.js)
- Language/framework files (.sass_cache, npm-debug.log)
- Files downloaded with package managers (node_modules)
- Credentials/tokens (wp-config.php)



gitignore.io

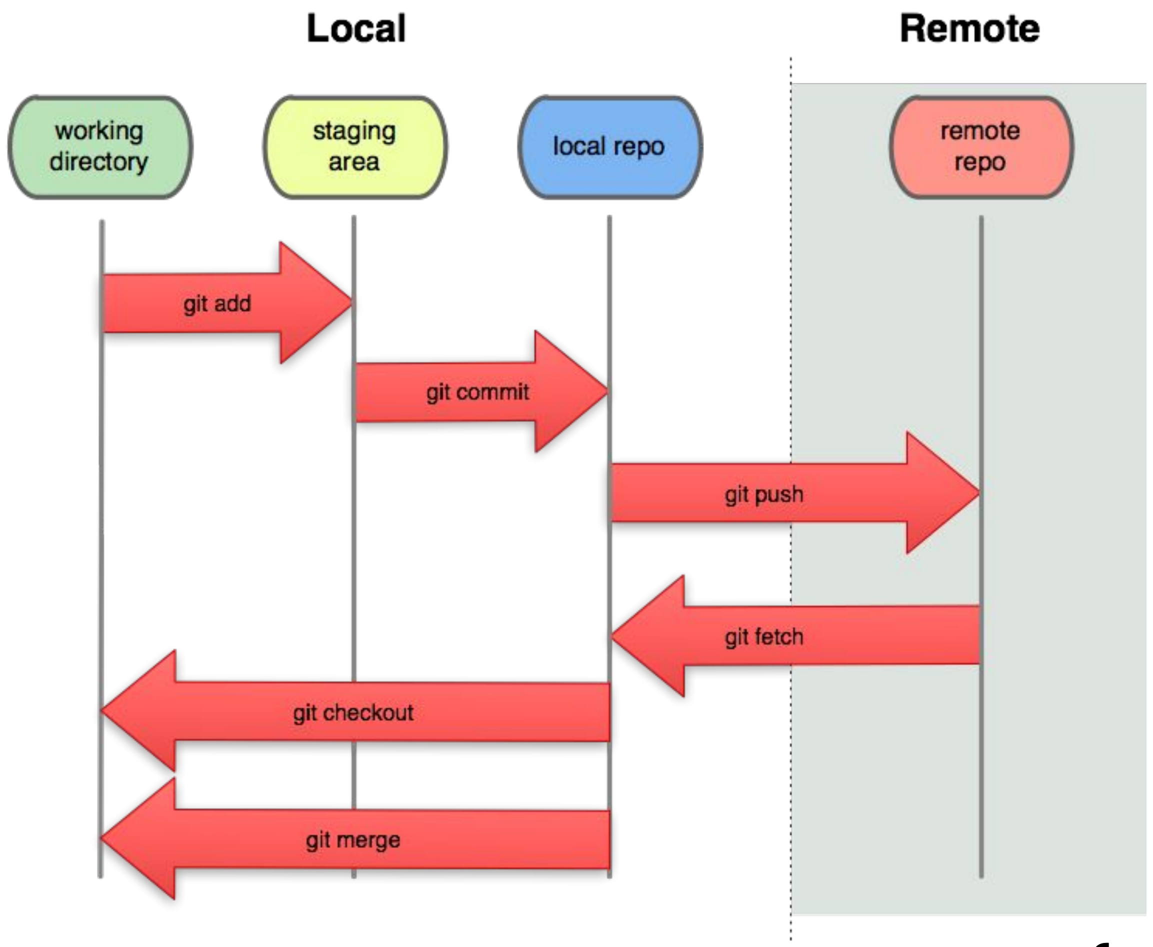
Create useful .gitignore files for your project

gitignore.io - Create Use
Vyacheslav
Secure
https://www.gitignore.io
Search Operating Systems, IDEs, or Programming Languages
Create
Source Code | Command Line Docs | Watch Video Tutorial

softserve

Basic git data transport commands

- `git add`
- `git commit`
- `git push`
- `git fetch`
- `git checkout`
- `git merge`



softserve

Additional important commands

Get help:

- **`git help <command>`**
- **`git <command> --help`**

Show status and log:

- **`git status`** – Show the working tree status
- **`git log`** – Show commit logs
- **`git ls-files -s`** - Show files in the index

Remove and revert:

- **`git rm`** – Remove files from the working tree and from the index
- **`git reset`** - Resets changes

Shortcuts:

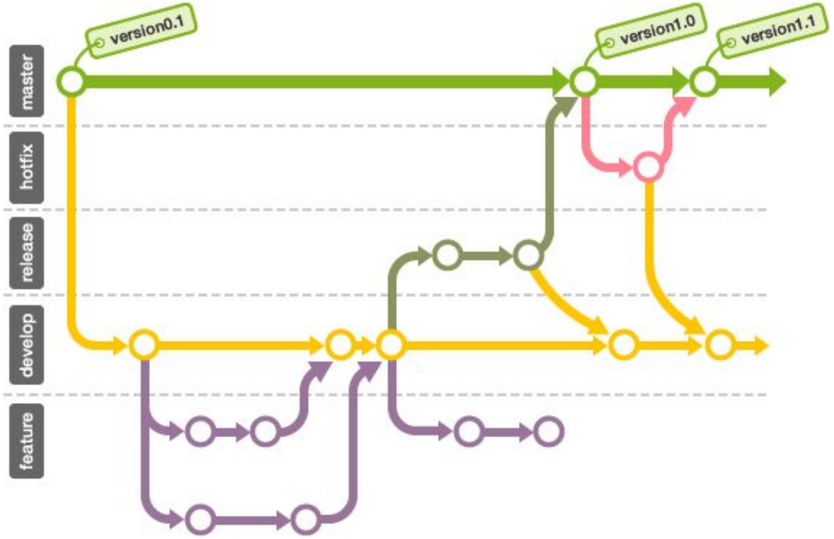
- **`git commit -am`** - combines `add` and `commit`
- **`git pull`** - Combines `fetch` and `merge`

Branches

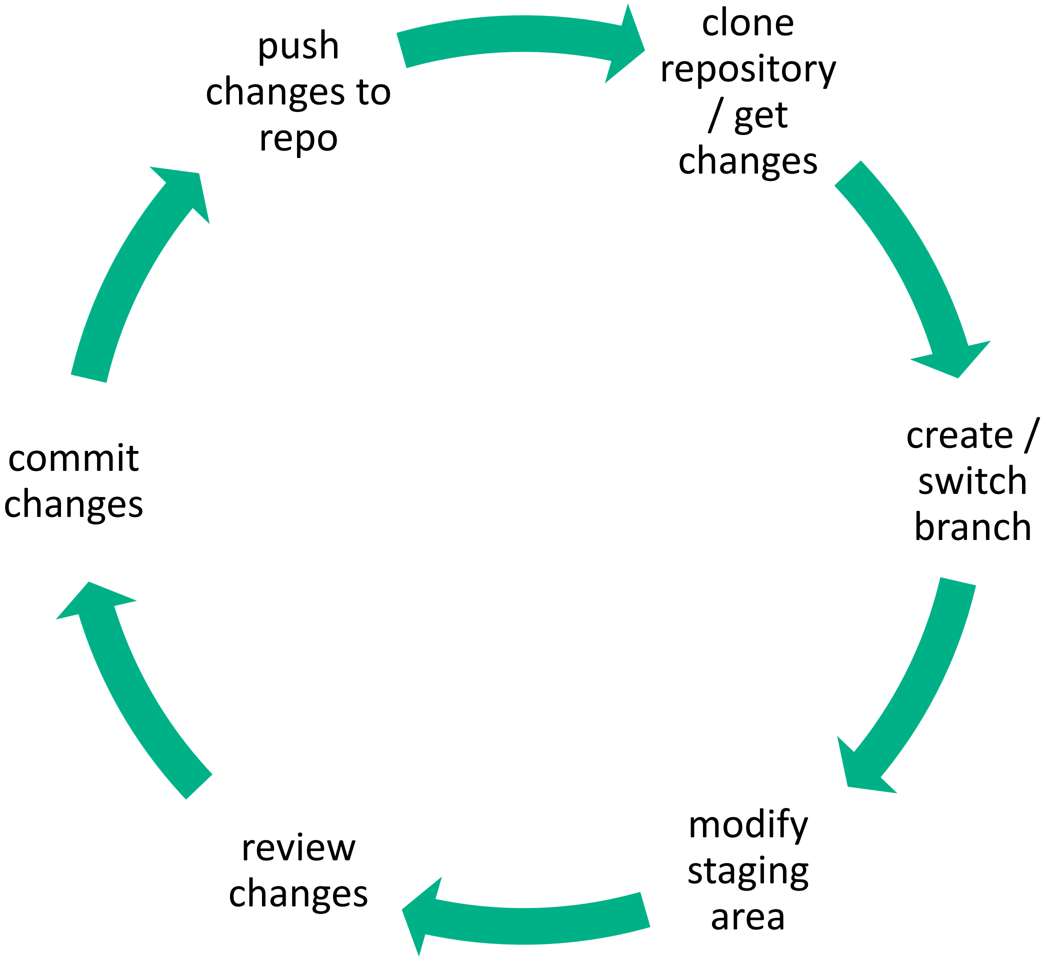
A **branch** represents an independent line of development.

Commands:

- `git branch` – list of branches in local repo
- `git branch <name>` – create new local branch named “name”
- `git branch –d <name>` – delete the branch named “name”
- `git branch –m <name>` – rename the current branch to “name”



Workflow



Clone repository

- `git clone`
- `git init`

Create/switch branch

- `git branch`
- `git checkout`

Add files to staging area

- `git add`

Review/merge changes

- `git status`
- `git log`
- `git diff`
- `git merge`

Commit changes

- `git commit`

Push changes to repo

- `git push`

Get changes from remote repo

- `git fetch`
- `git pull`

softserve

Recommended links

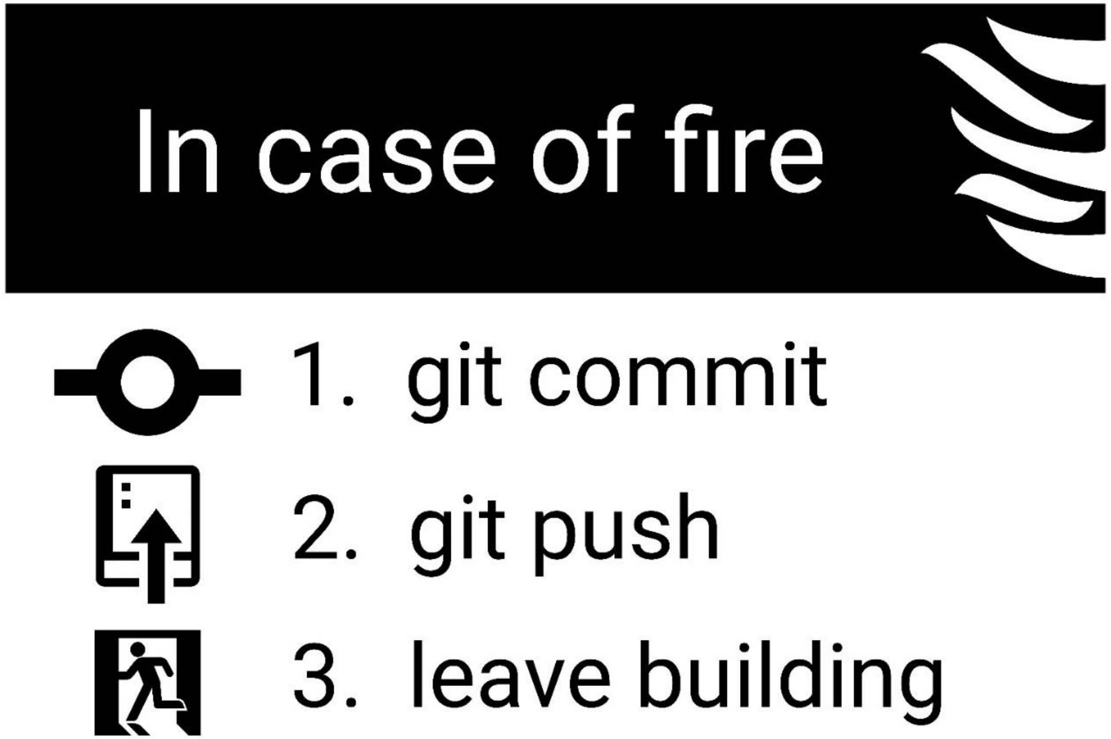
https://git-scm.com/book/en/v2 - original documentation from Git team

https://www.atlassian.com/git/tutorials - Atlassian git tutorial

https://try.github.io - git course from codeschool

https://learngitbranching.js.org/ - practical course on git branching

Thank you!



In case of fire

1. git commit
2. git push
3. leave building

softserve

by Vyacheslav Koldovskyy