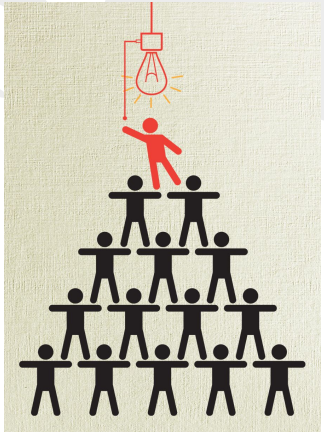
Introduction

Git and Github

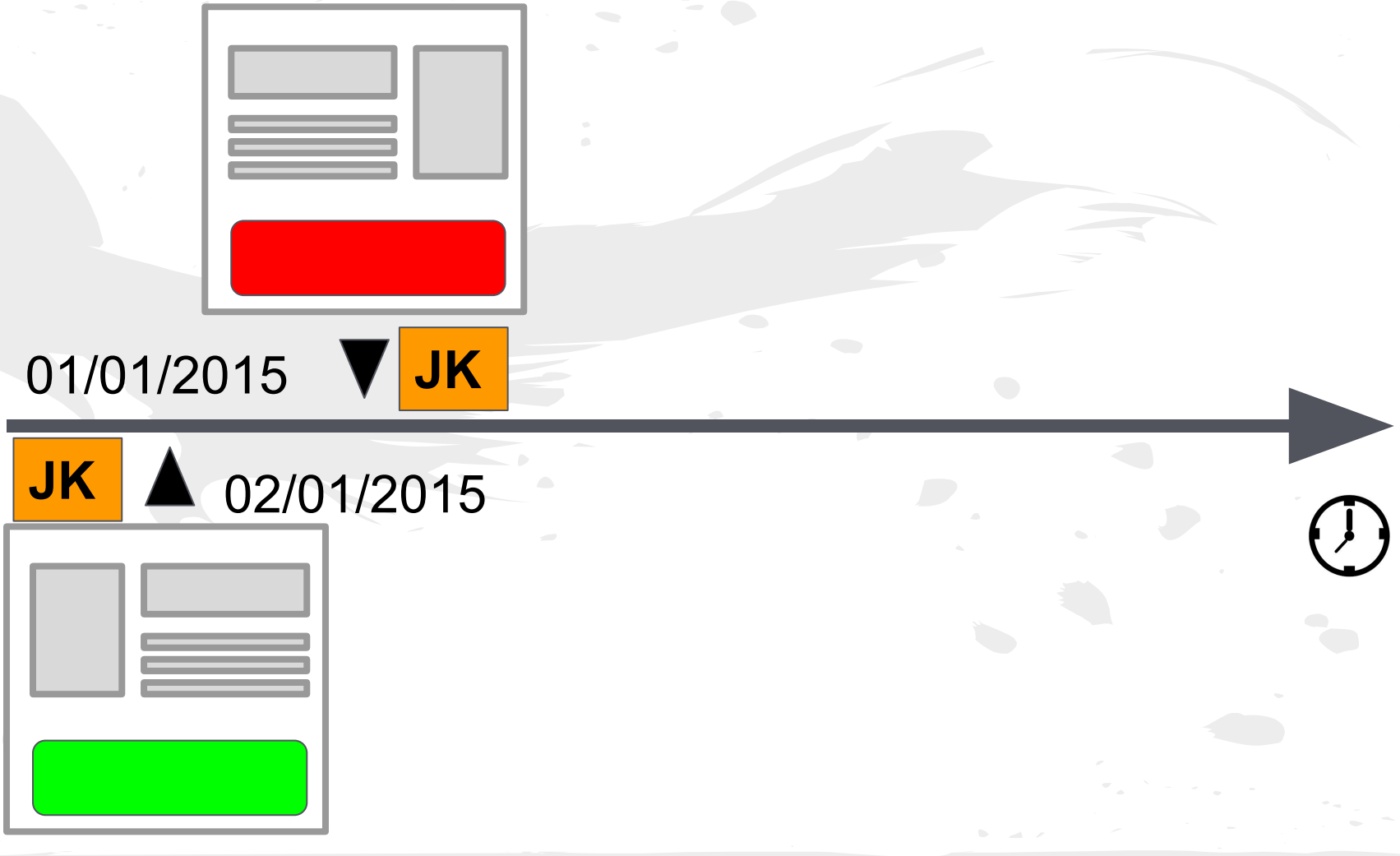
git

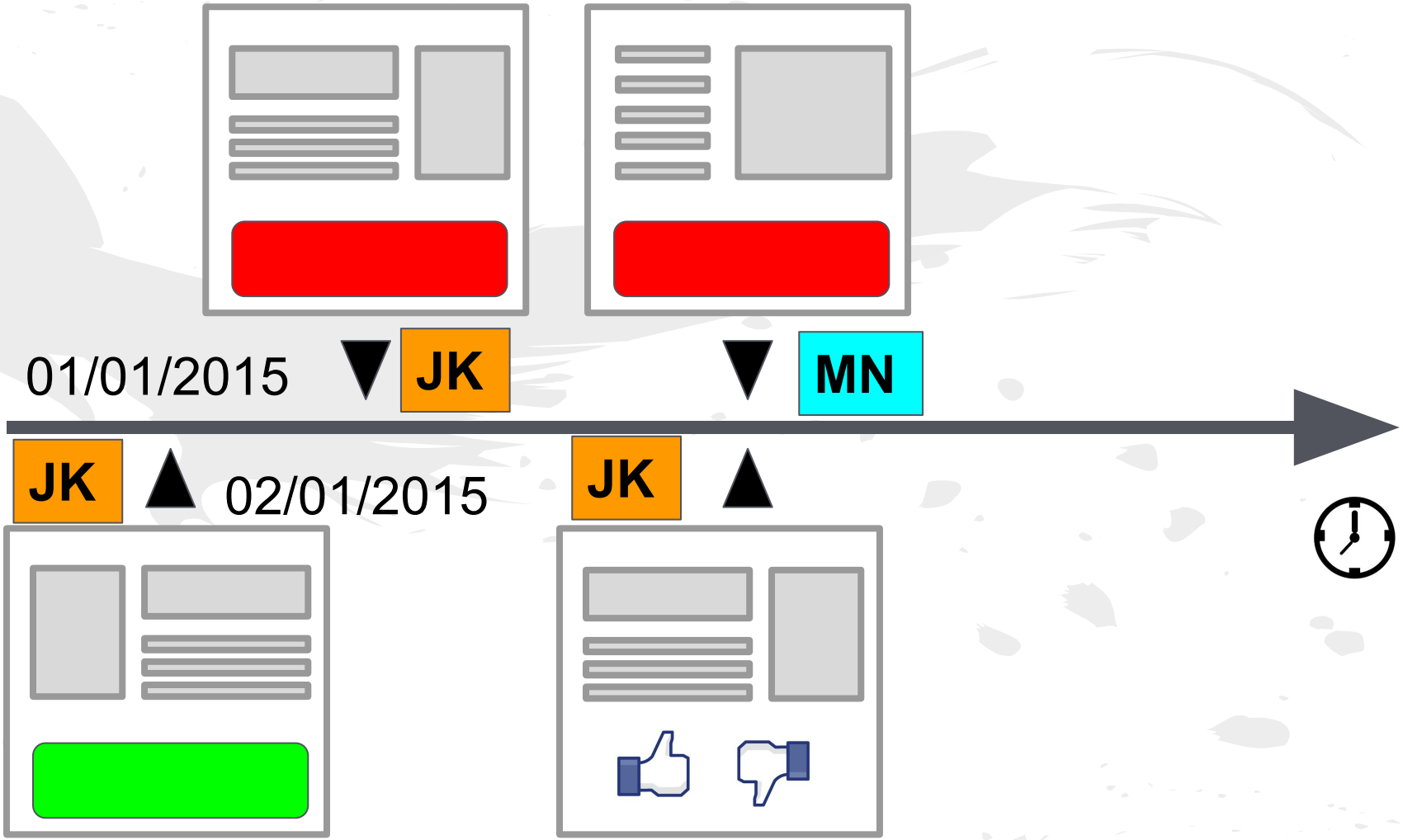
Version Control

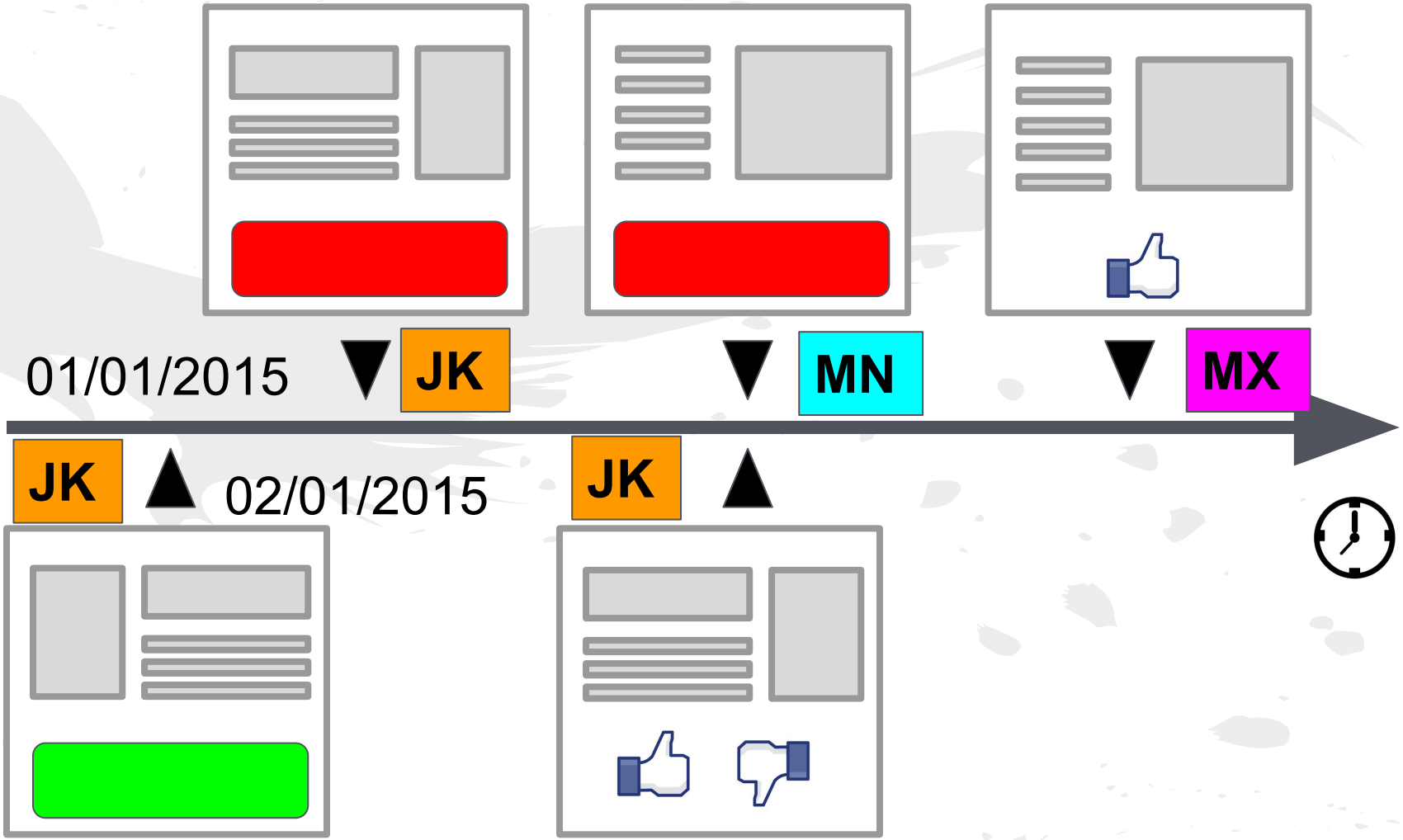

Ctrl
Z



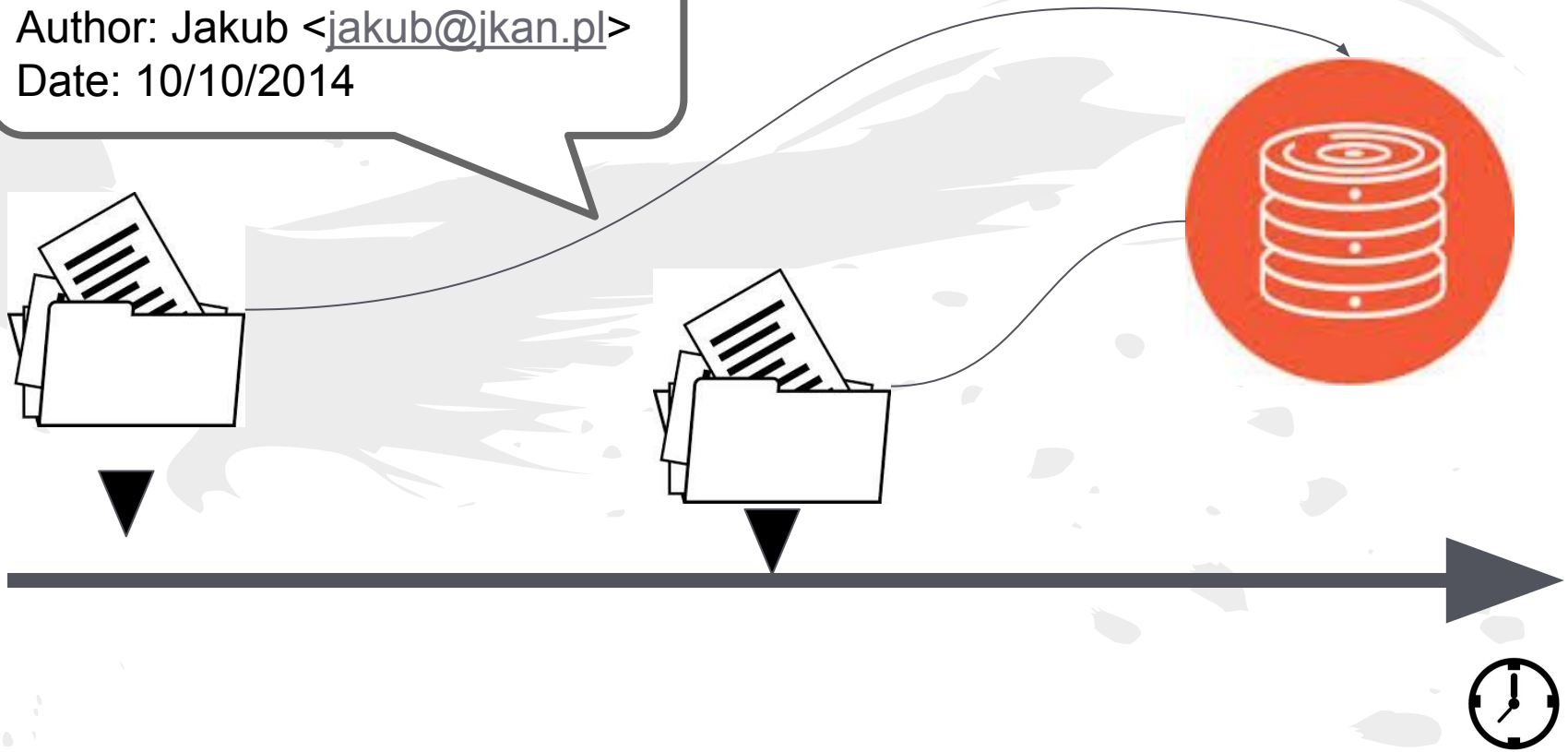
01/01/2015


01/01/2015
JK
JK
02/01/2015

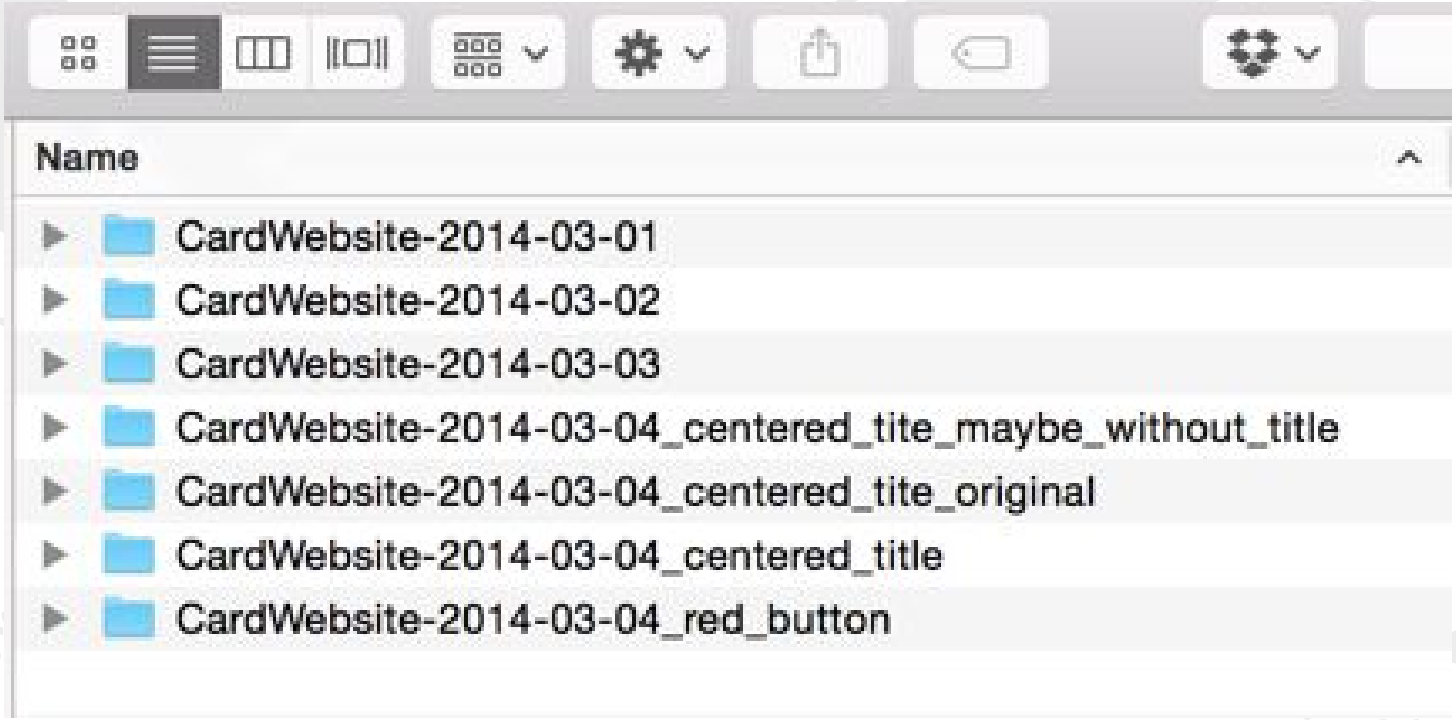

01/01/2015
JK
MN
JK
02/01/2015
JK


01/01/2015
JK
MN
MX
JK
02/01/2015
JK

Remove dislike button
Author: Jakub <jakub@jkan.pl>
Date: 10/10/2014

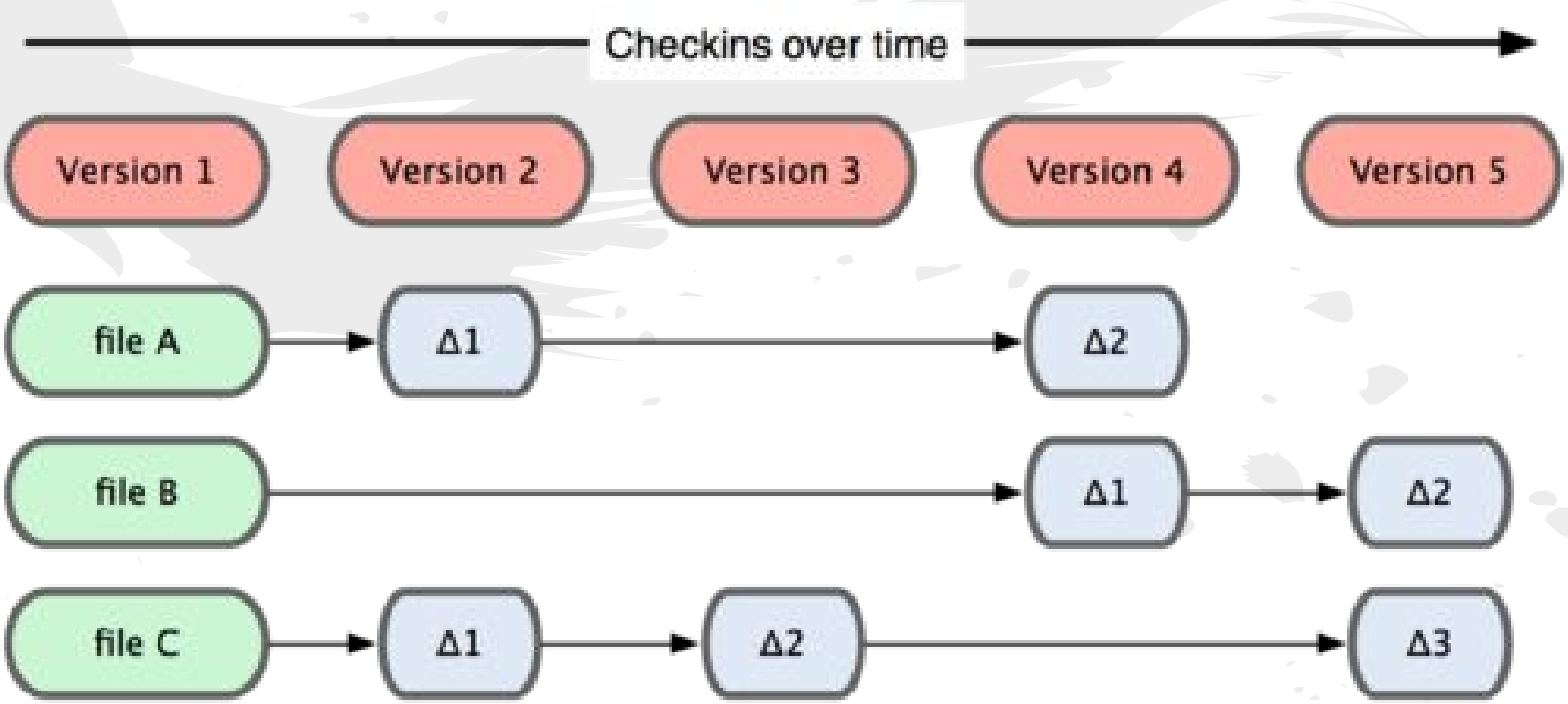


Why Version Control

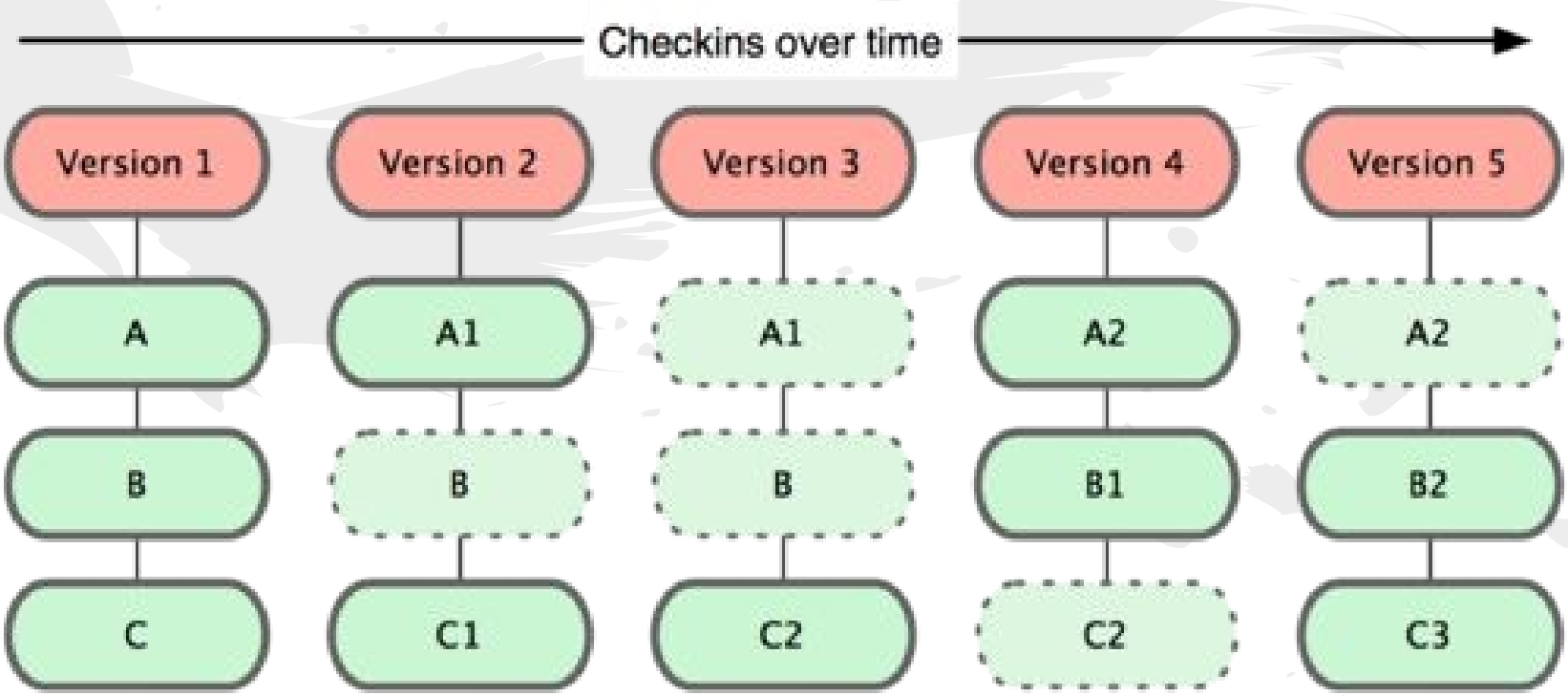


Not only source code

Why Version Control



Why Version Control



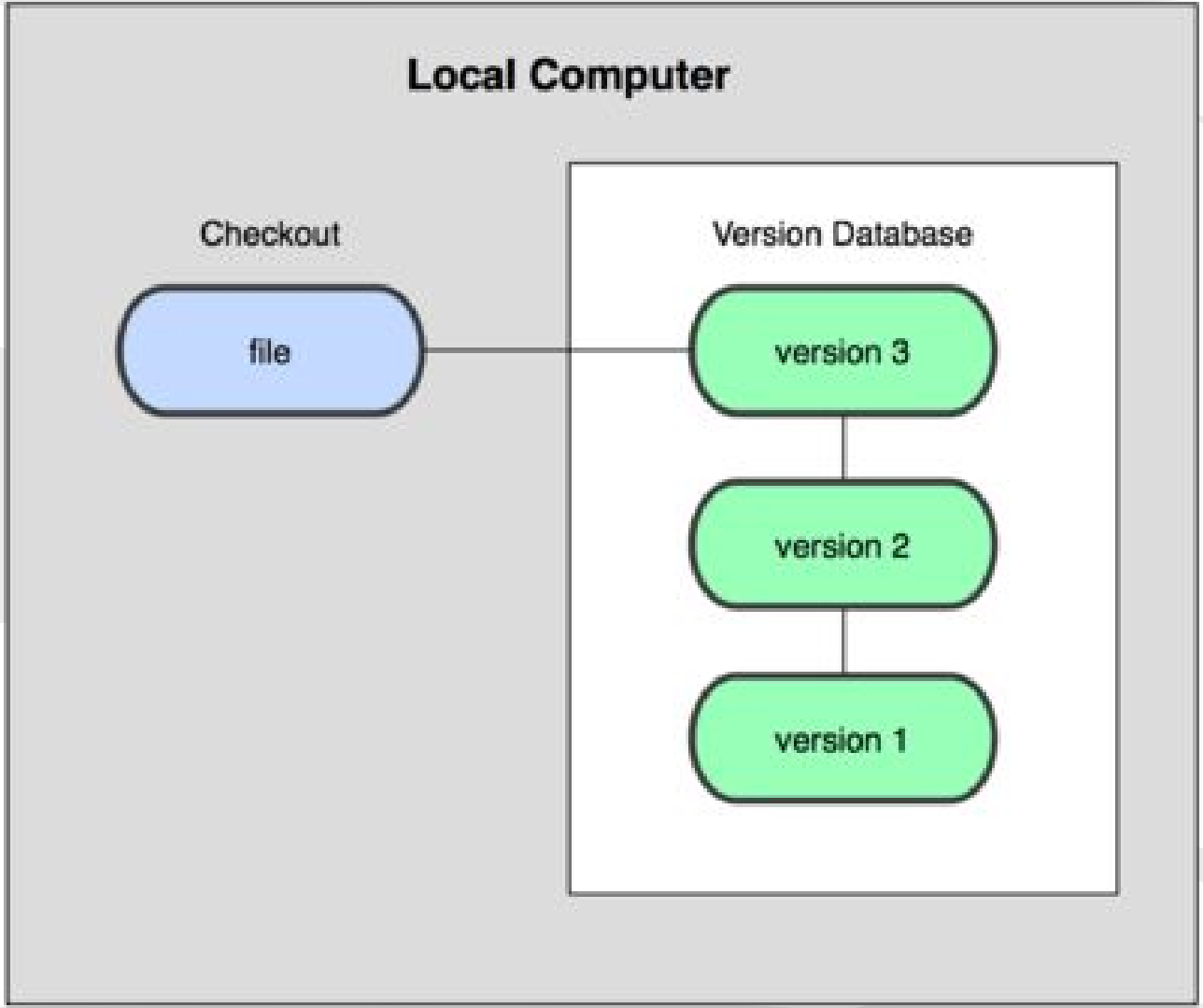
Why Version Control

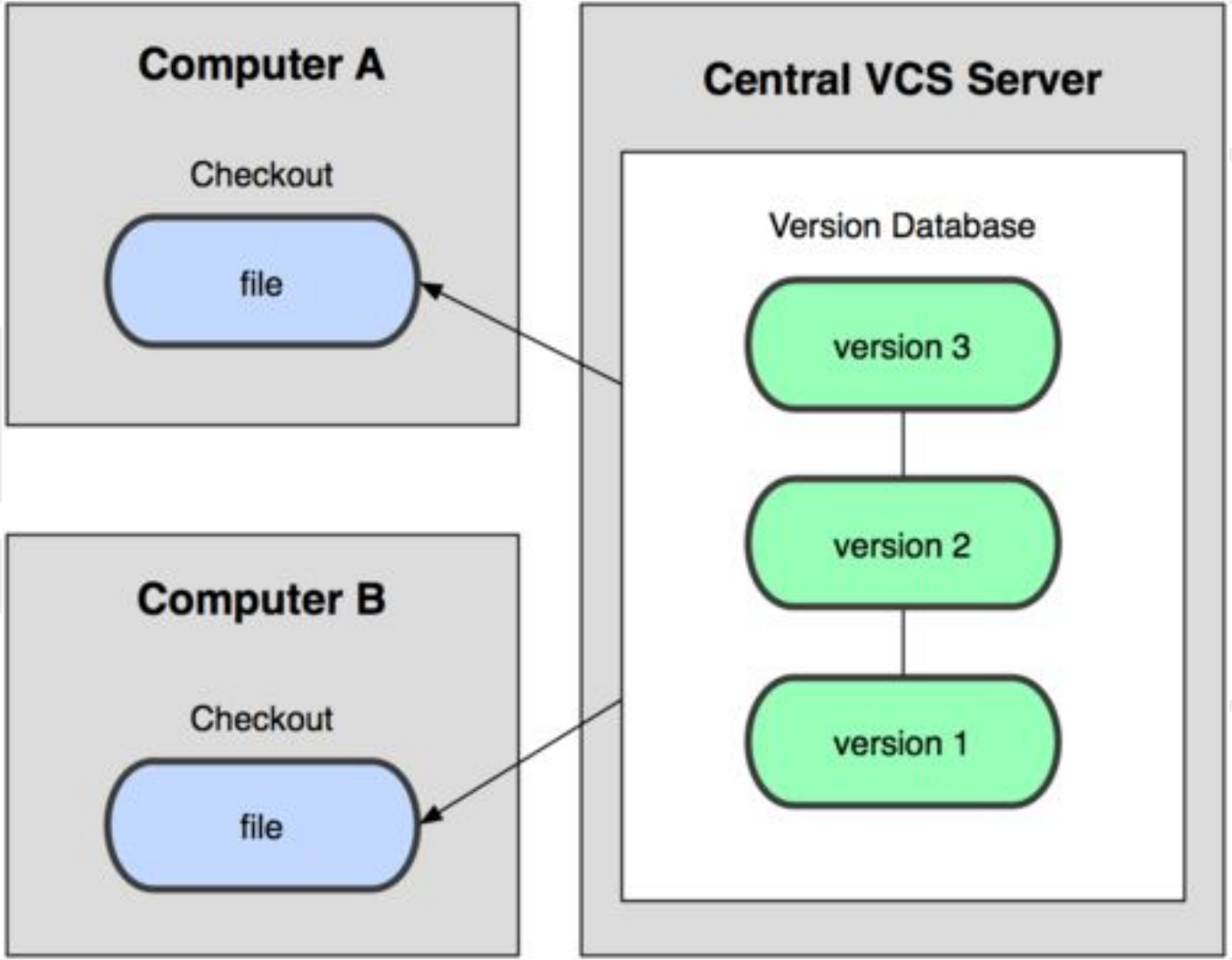
- Save versions properly
- Understand what happend
- Resorte previous versions
- Collaborate without overwritting
- Backup as side benefit

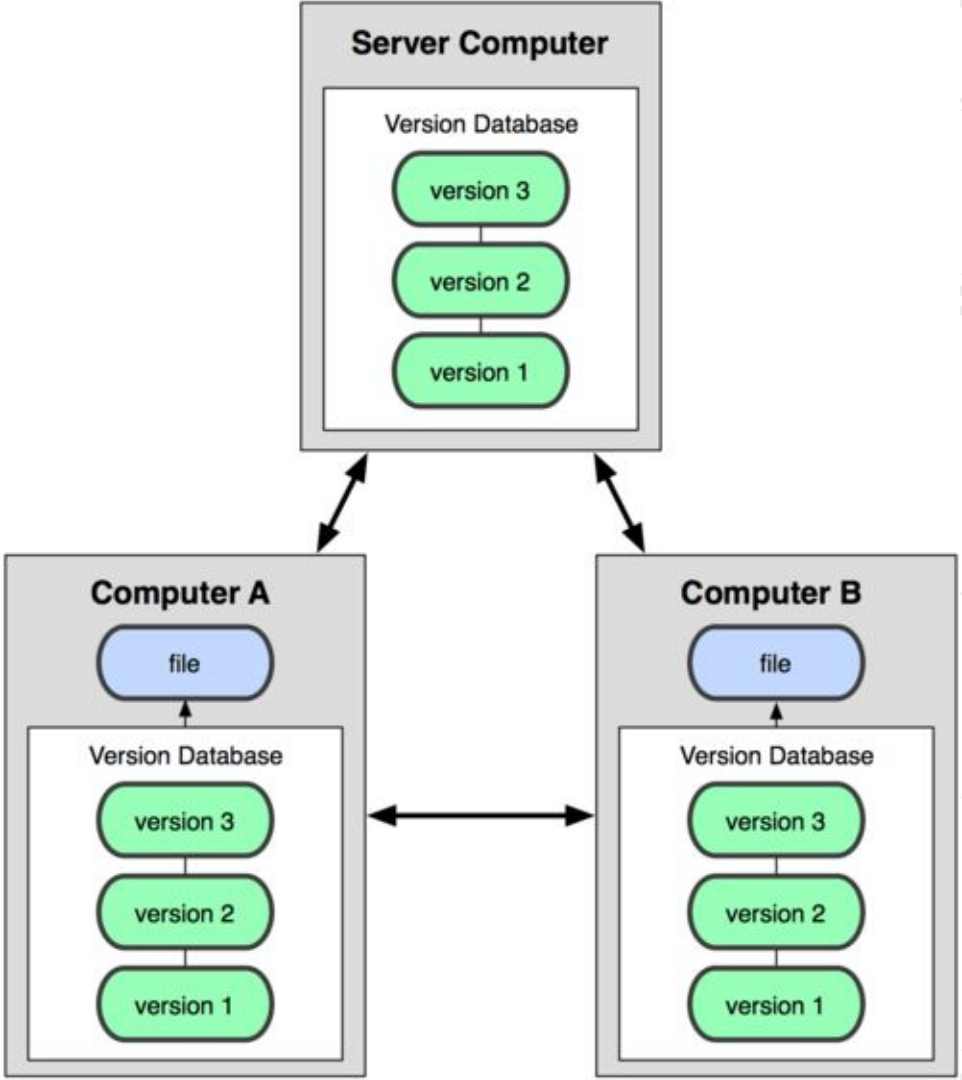
Version Control - GIT

- Independent of **OS**
- independent of tools (IDE, editor)
- independent of **language / framework**

Version Control - types


Local Computer
Checkout
file
Version Database
version 3
version 2
version 1


Computer A
Checkout
file
Computer B
Checkout
file
Central VCS Server
Version Database
version 3
version 2
version 1


Server Computer
Version Database
version 3
version 2
version 1
Computer A
file
Version Database
version 3
version 2
version 1
Computer B
file
Version Database
version 3
version 2
version 1

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.

Linus Torvalds 7 April 2005

git config

- interface to set up git environment
- typically only need to use on a new machine

```
git config <options>
```

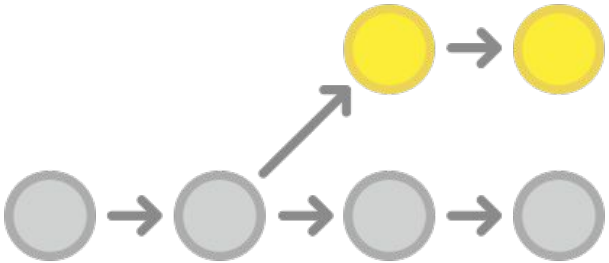
git config

```
git config --global user.name “Jakub Kanclerz”
```

```
git config --global user.email “jakub@jkan.pl”
```

Git Basics

Setting up a Git Repository



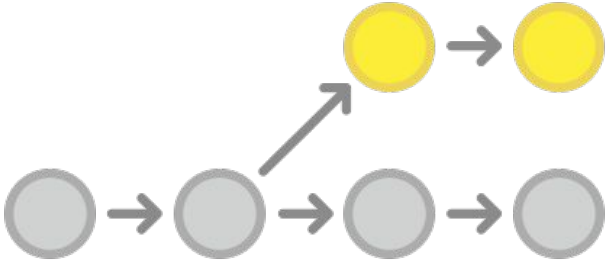
git init

- the command Creates a new Git repository.
- convert an existing project


```
git init <directory>
```

Git Basics

Recording snapshots



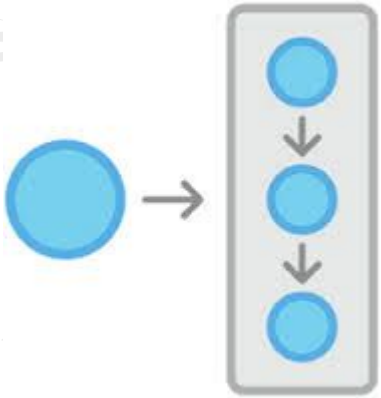
git status

- displays the state of the working directory

```
git status
```

git add

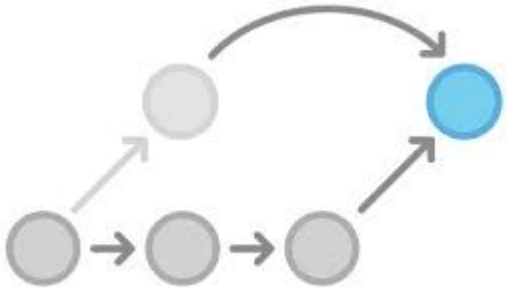
- moves changes to staging area
- opportunity to prepare transaction befor committing it to history



```
git add <file/directory>
```

git commit

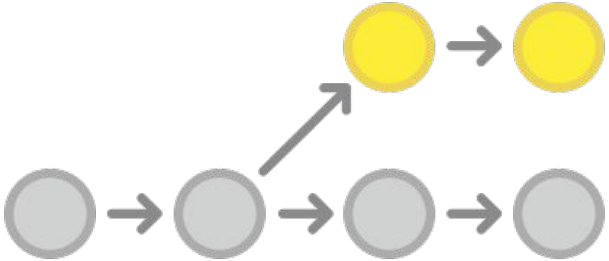
- takes staged snapshot
- commits it to the project history
- Combined with git add, defines basic workflow



```
git commit -m <message>
```

Git Basics

Inspecting a Git repository



git log

- lets to explore the previous revisions of a project

```
git log <options>
```

git diff

- lets to explore the previous revisions of a project
- Show changes between commits, commit and working tree

```
git diff <commit>
```

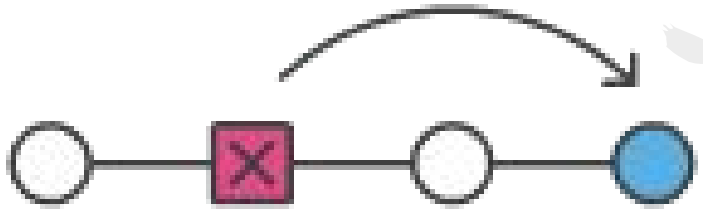
git blame

- Lets see who put it into codebase

```
git blame -L 15,16 foo/bar.txt
```

Switch between snapshots

Viewing old commits



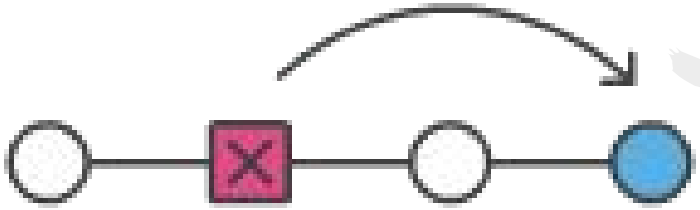
git checkout

- checking out filles
- checking out commits
- checking out branches

```
git checkout <commit/branch>
```

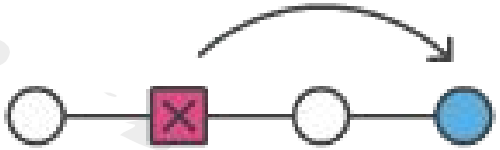
Undoing Changes

back to old commits



git revert

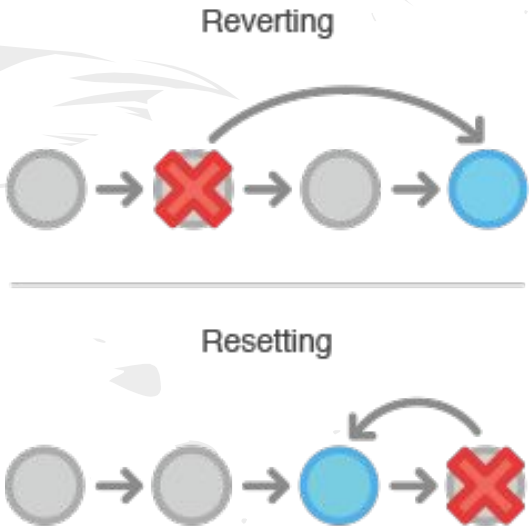
- undoes a committed snapshot
- appends a new commit with the results



```
git revert <commit>
```

git reset

- undoes changes to files in the working directory
- lets clean changes that have not been pushed to a repository




```
git reset <options> <commit/file>
```

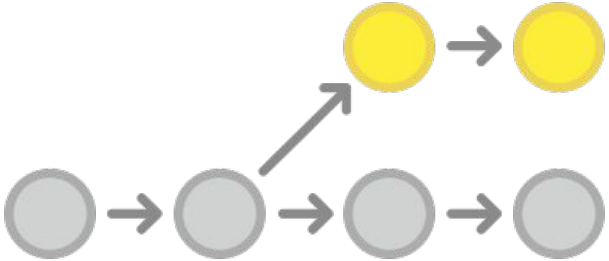
git clean

- removes untracked files

```
git clean <options>
```

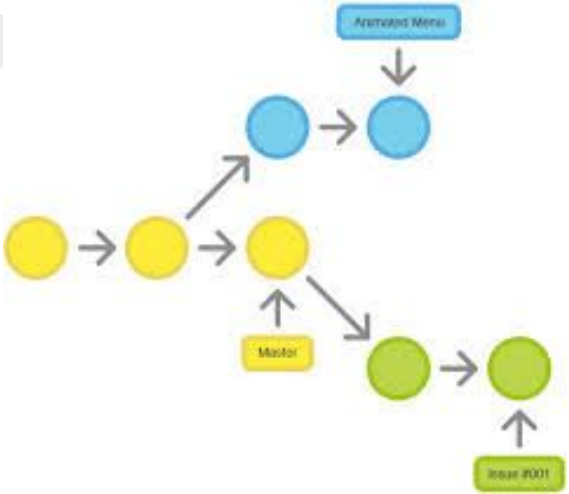
Git Branches

Git branches



git branch

- branch administration tool



```
git branch <branch>
```

git checkout

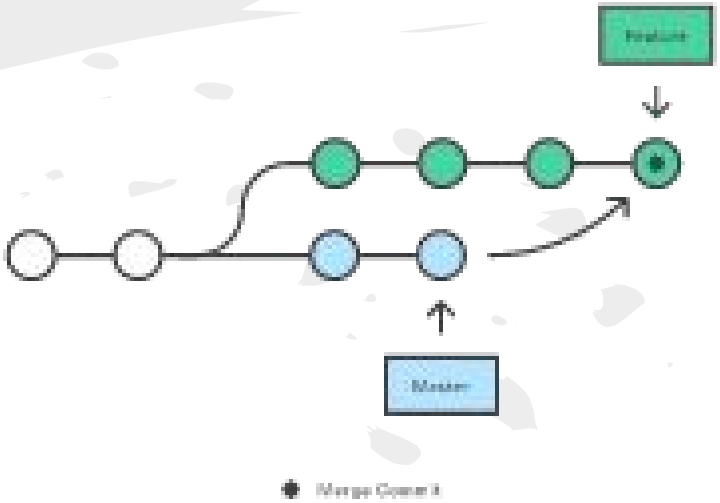
- checking out filles
- checking out commits
- checking out branches

```
git checkout -b new_branch_name
```

```
git checkout <commit/branch>
```

git merge

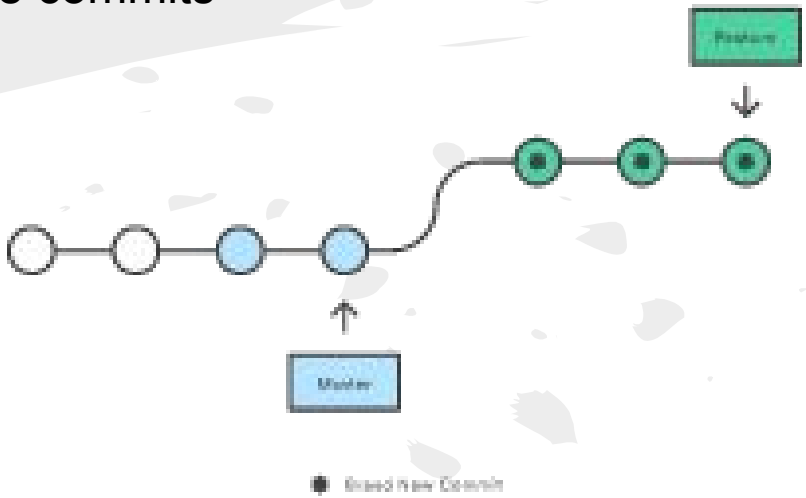
- integrates changes from branches



```
git merge <options> <branch>
```

git rebase

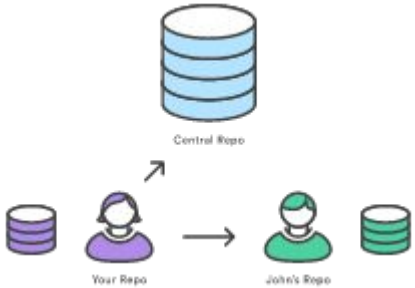
- hepls avoid unnecessary merge commits



```
git rebase <base>
```

Remote Synchronization

Remote Git repositories



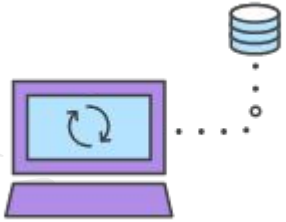
git remote

- administration tool for managing remote connections
- allows to define shortcut for repositories' urls

```
git remote <options>
```

git fetch

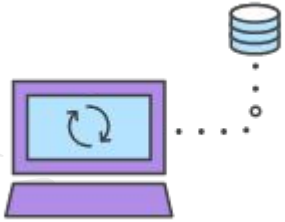
- allows downloads a branch from another repository



```
git fetch <remote> <branch>
```

git pull

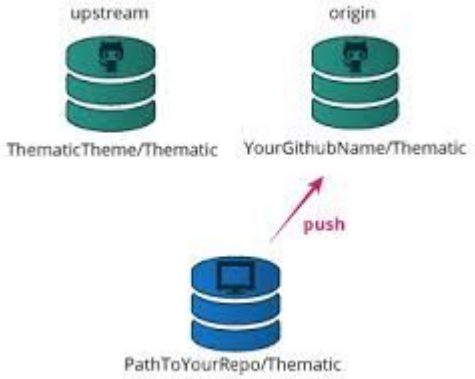
- allows downloads a branch from another repository, then immediately merges it into the current branch



```
git pull <options> <remote>
```

git push

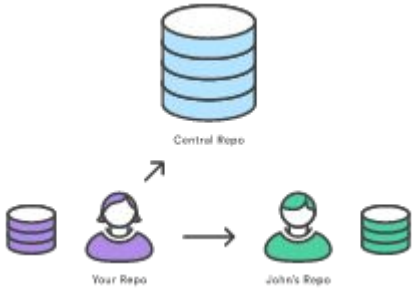
- lets move a local branch to anotcher repository



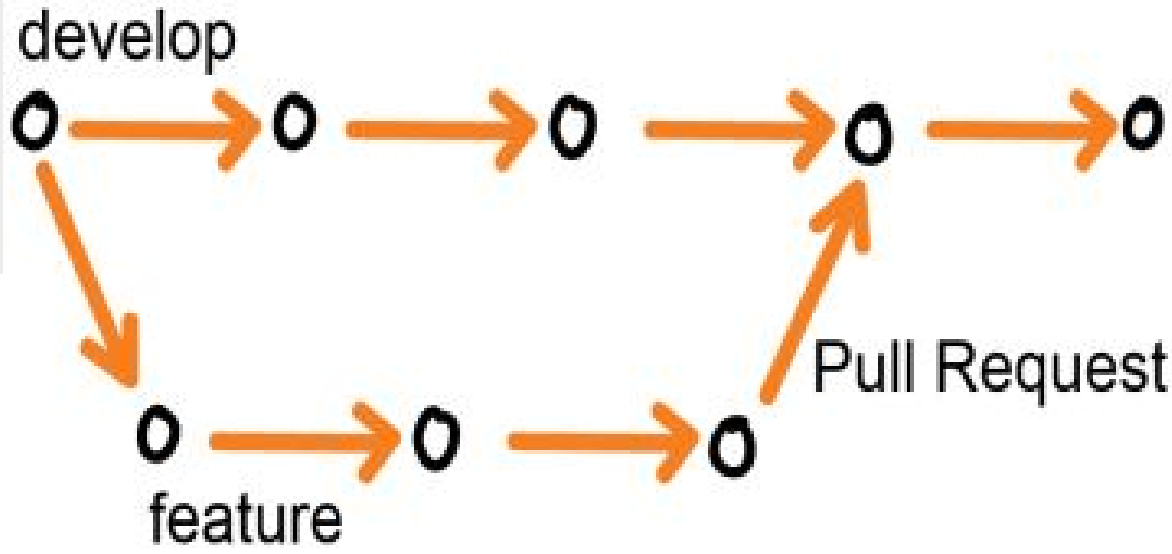

```
git push <remote> <branch>
```

Git Workflow

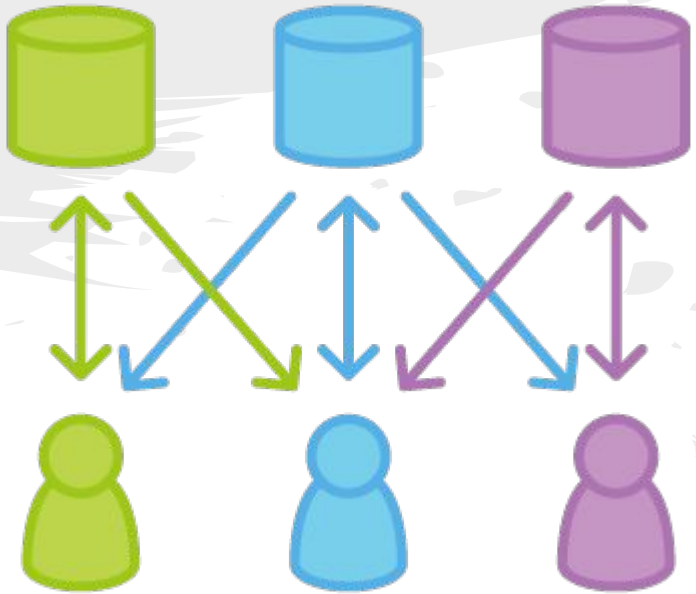
Centralised Workflow



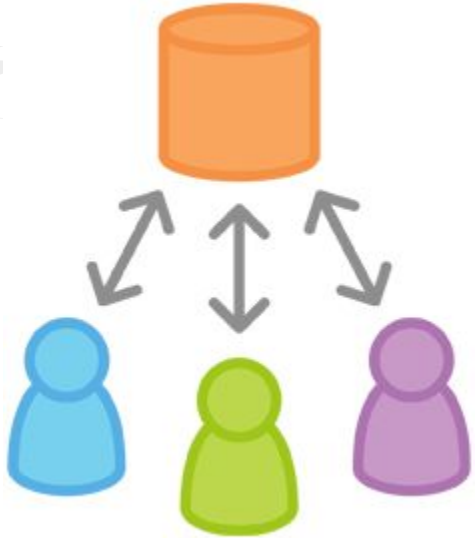
Feature Branch



Forking



Collaborating



git clone

- Creates copy of an existing repository


```
git clone <repo> <directory>
```