Git Basics

PV260 Software Quality

Stanislav Chren

23. 2. 2017

What is Git

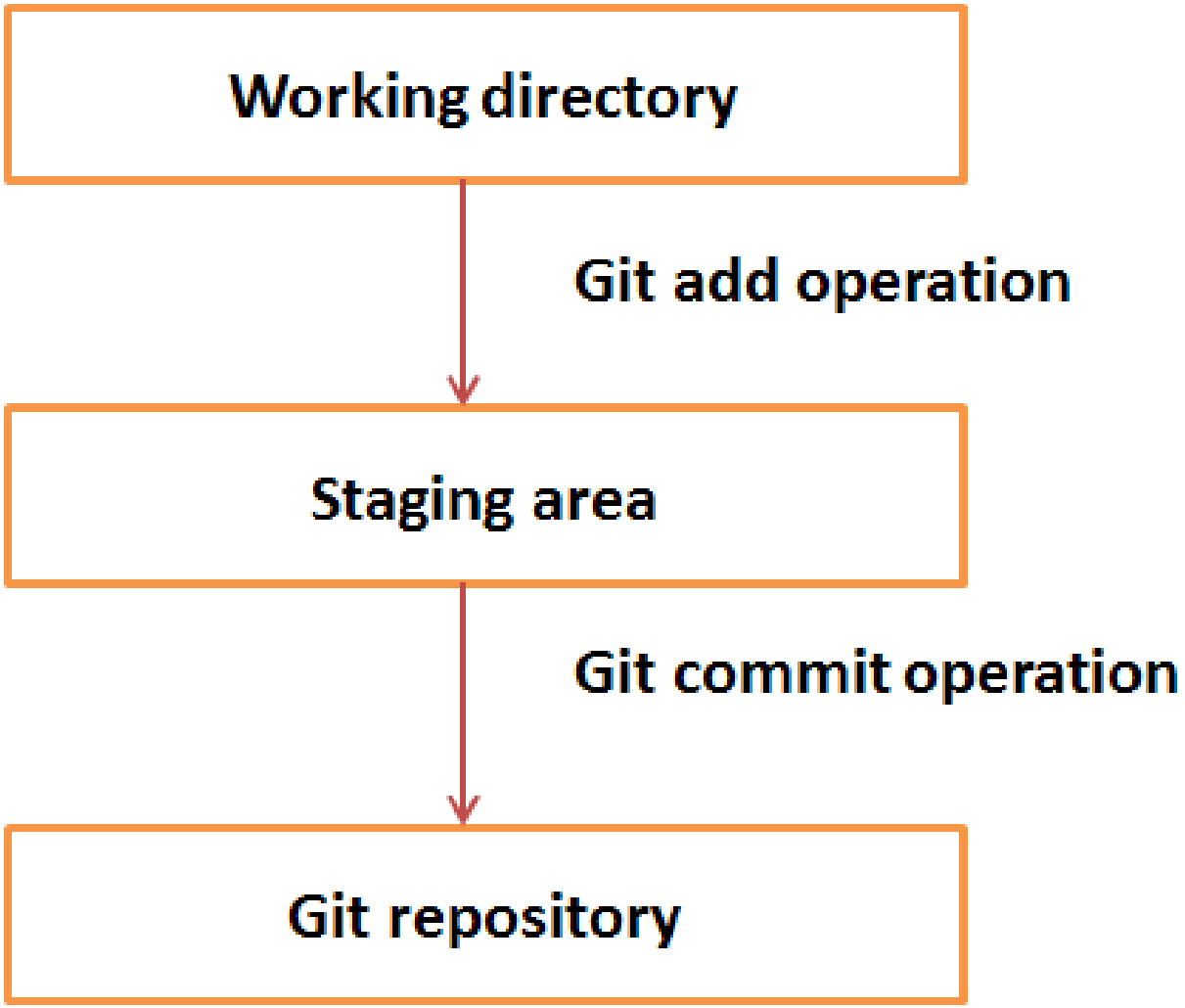
Git is a widely used source code management system for software development. It is a distributed revision control system with an emphasis on speed, data integrity, and support for distributed, non-linear workflows.

Our Use case

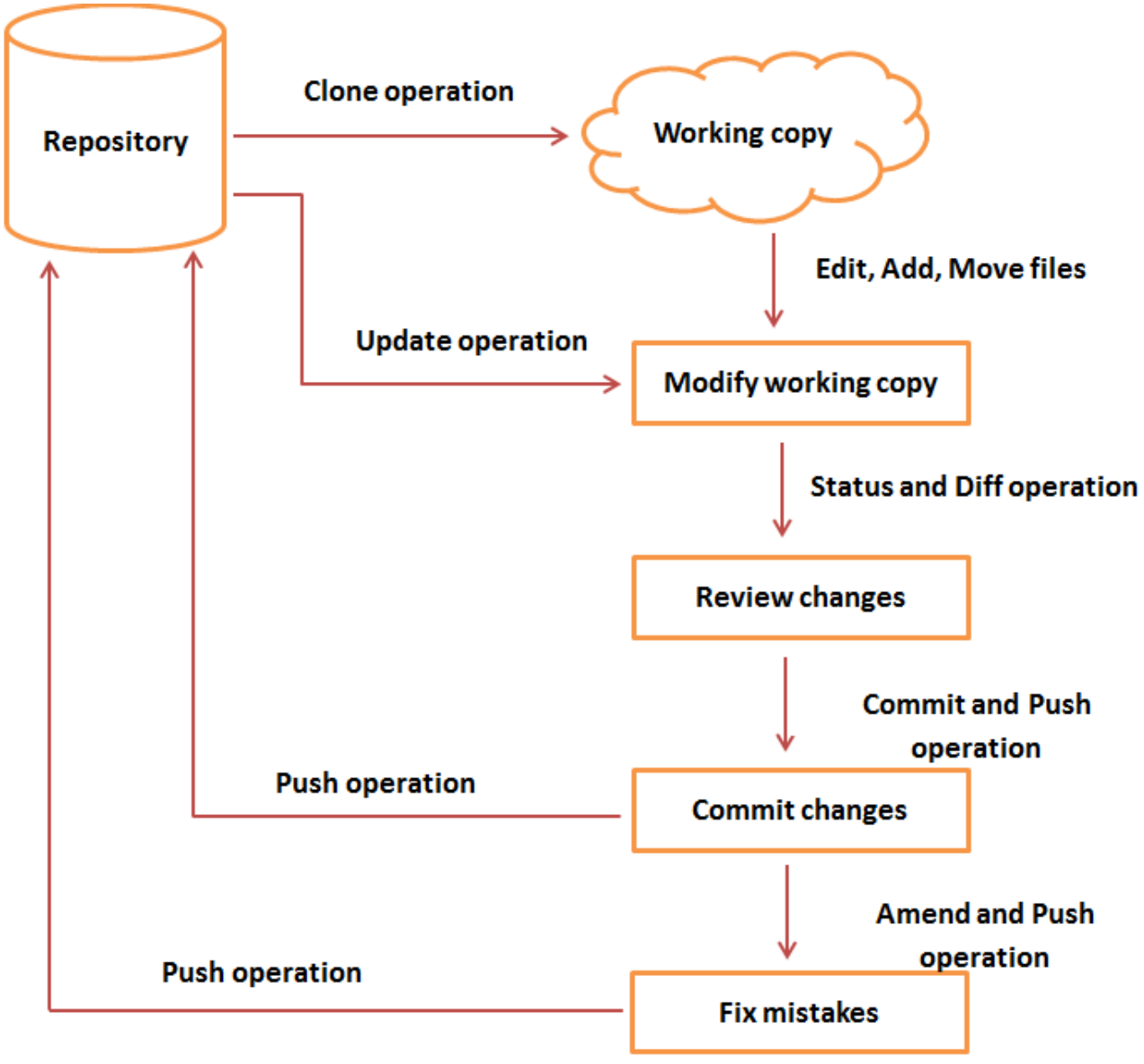
- Materials provided as repositories reflecting change in the code from start of refactoring / covering by tests etc. to the end of the process
- Students encouraged to use some SCM for their assignments

download from `https://git-scm.com/`

Basic Workflow



Git Life Cycle



Basic commands 1

- Commands are written without the '-' e.g. `git commit`
- Initial environment setup:

```
git config --global user.name "Your Name"
git config --global user.email "your@email.com"
```

Command	Description
help	Prints help summary, all commands are described here
init	Creates a new repository in the current folder
clone	Creates an instance of the remote repository.

Basic commands 2

Command	Description
add	Add file contents to the index
rm	Remove files from the working tree and from the index
commit	Record changes to the repository
push	Update remote repository
pull	Fetch from and integrate with another repository or a local branch
fetch	Download objects and refs from another repository
checkout	Switch branches or restore working tree files
branch	List, create, or delete branches

Basic commands 3

Command	Description
status	Shows the working tree status
log	Shows commit logs
show	Displays detailed information about various objects (blobs, trees, tags and commits)
diff	Shows changes between commits, commit and working tree, etc

Creating a Repository

- Create an empty reopsitory in the current directory
  - `git init`
  - add `--bare` option to convert current dir into repository. The directory name should have form *name.git*
- Clone an existing repository into a newly created directory in the current folder.
  - `git clone PATH_TO_REPO`
  - `PATH_TO_REPO` can be path in local filesystem or URL

Add

- Updates the index using the current content found in the working tree, adds files to the staging area
- `git add OPTIONS`

Option	Description
FILE	Stages the given FILE
`--update, -u`	Stage modified and deleted files only
`--all, -A`	Stage all (new, modified, deleted) files

Commit

- Stores the current contents of the index/staging area in a new commit along with a log message from the user describing the changes.
- New files need to be added by `git add` first
- `git commit OPTIONS`

Option	Description
`-m MESSAGE`	Stores the changes with the given MESSAGE. Mandatory
`--all, -a`	Automatically stages files that have been modified and deleted, but new files you have not told Git about are not affected.
`--dry-run`	Does not execute the commit, just prints a summmary of what is to be commited

- The commited changes can be stored in remote repo by: `git push`

Pull

- Incorporates changes from a remote repository into the current branch.
- In its default mode, `git pull` is shorthand for `git fetch` followed by `git merge FETCH_HEAD`

Status

- Displays paths that have differences between the index file and the current HEAD commit, paths that have differences between the working tree and the index file, and paths in the working tree that are not tracked by Git (and are not ignored
- The symbols beside filename represents a state of the file, e.g. `??` for untracked file, `A` for added, `M` for modified etc.
- `git status OPTIONS`

Option	Description
`--short, -s`	Give the output in the short-format.
`--long, -a`	Give the output in the long-format. Default.
`--ignored`	Show ignored files as well.
`--column`	Display untracked files in columns.

Log

- Shows the commit logs.
- Each entry contains the SHA-1 checksum, the author's name and email, the date written, and the commit message.

Option	Description
`-p`	Shows the difference introduced in each commit
`--stat`	Shows abbreviated stats for each commit
`--pretty=STYLE`	Changes output format based on the STYLE. Posibble values - oneline, full, fuller, short
`--column`	Display untracked files in columns.

Diff

- Shows changes between the index and a tree, changes between two trees, changes between between two files on disk.
- Compare working tree and index
  - `git diff`
- Compare two revisions
  - `git diff REV1 REV2`
- Compare two files from different revisions
  - `git diff REV1:file REV2:file`

Tutorials

- http://www.tutorialspoint.com/git/index.htm
- https://www.atlassian.com/git/tutorials/
- http://git-scm.com/docs/gittutorial

Task 1

1. Create two new repositories, named *repo1* and *repo2*
2. For the former, use `git init` command, for the latter use also the `--bare` option. What is the difference?
3. Create a github[1]/gitlab[2]/bitbucket[3] account (if you don't have it already)
4. Clone a repository PV260 from `https://github.com/stanozm/PV260.git`

[1] https://github.com/
[2] https://gitlab.fi.muni.cz/
[3] https://bitbucket.org/

Task 2

1. What is the content of the 255471.hello file after 3rd revision?
2. What changes were made to the file in 2nd revision?
3. When was the file 255471.bye created and removed from the repo?
4. What was its content?

Task 3

1. Create two files with names `YOUR_UCO.hello` and `YOUR_UCO.bye`
2. Tell the repository to start tracking the new files
3. Push them to the remote repository
4. Edit the .hello file so that it contains your name and year of your studies.
5. Commit and push the changes.
6. Delete the .bye file so that it is no longer tracked.