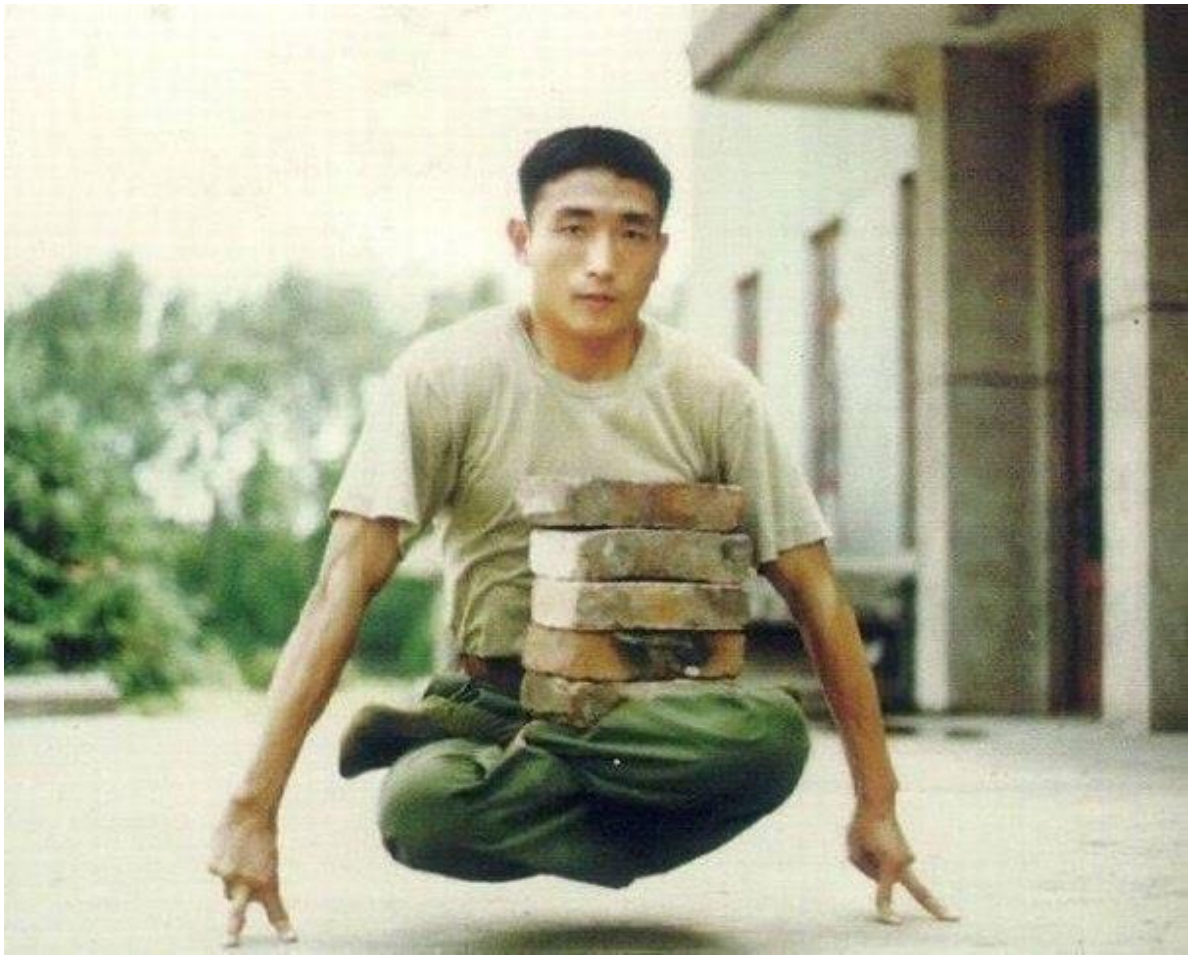
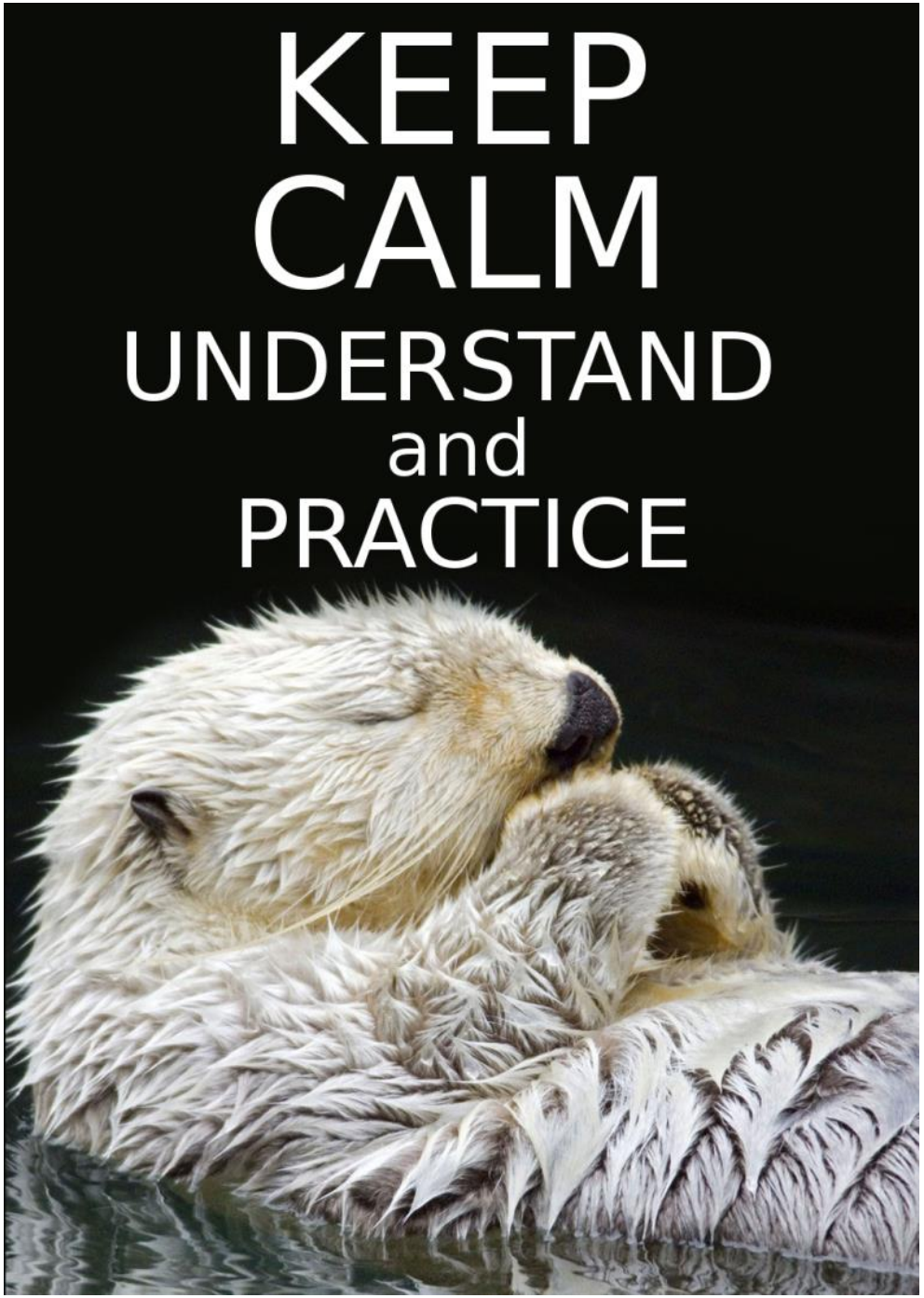
Intro to **Git**

Vladimir Alekseichenko

Good practices make life easier

Good practices are not easy




KEEP
CALM
UNDERSTAND
and
PRACTICE

Agenda

- History (RCS, VCS, SVN, TFS)
- Git
  - Getting and creating projects
  - Basic snapshotting
  - Branching and merging
  - Sharing and updating projects

Revision control system

It is a software implementation of revision control that automates the storing, retrieval, logging, identification, and merging of revisions.

RCS was first released in 1982 by Walter F. Tichy.

http://en.wikipedia.org/wiki/Revision_Control_System

Concurrent versions system

It is a client-server free software revision control system in the field of software development.

It also known as the Concurrent Versioning System.

http://en.wikipedia.org/wiki/Concurrent_Versions_System

Apache Subversion

It is a software versioning and revision control system distributed.

http://en.wikipedia.org/wiki/Apache_Subversion

TFS – what is it?

http://siriusbuzz.com/what-happens-with-that-sirius-lifetime-subscription.php/big-question-mark

TFS

Team Frustration Server

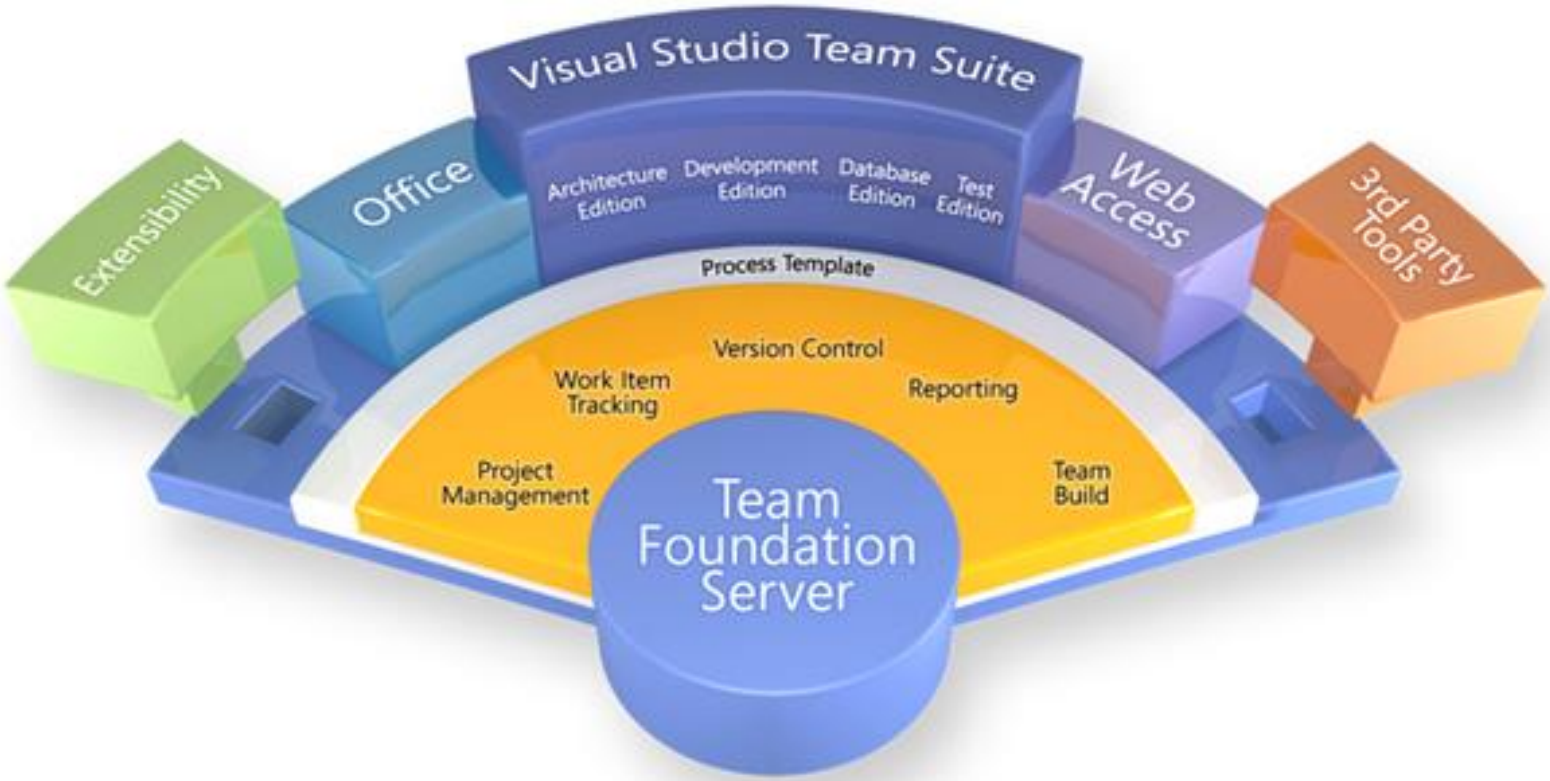
TFS

Tottaly Fucking Stupid

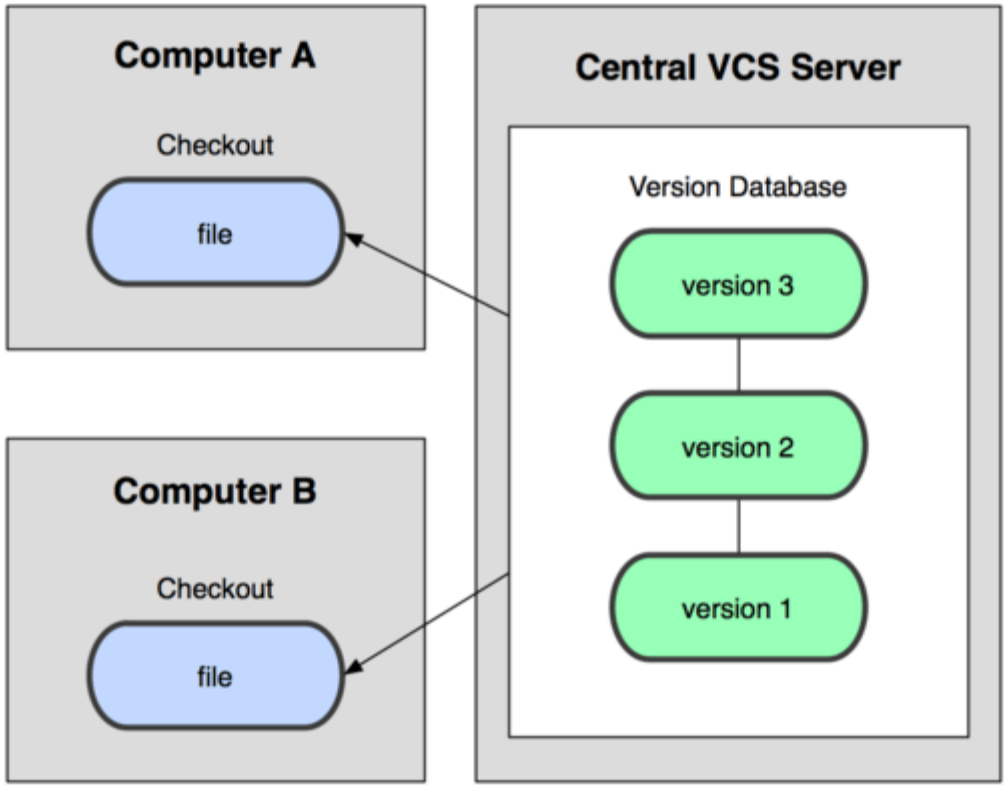
TFS

Tempt Fate & Suffer

TFS

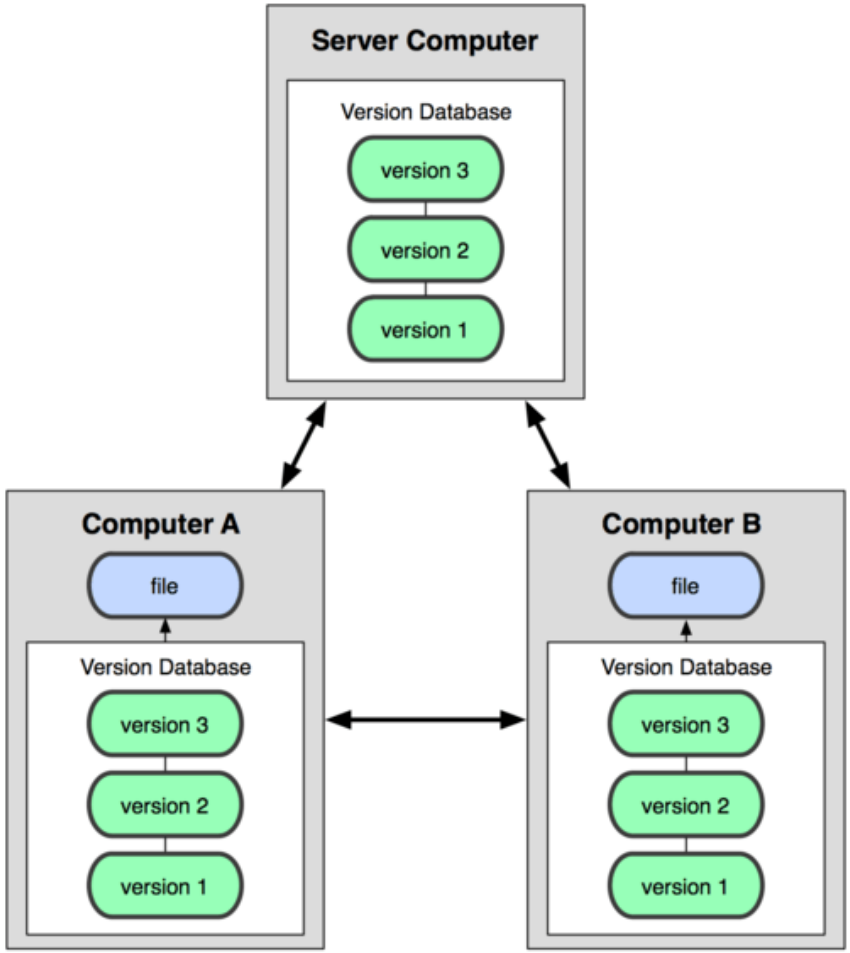

Visual Studio Team Suite
Extensibility
Office
Architecture Edition
Development Edition
Database Edition
Test Edition
Web Access
3rd Party Tools
Process Template
Version Control
Work Item Tracking
Reporting
Project Management
Team Build
Team Foundation Server

Centralized version control systems



http://git-scm.com/book/en/Getting-Started-About-Version-Control

Distributed version control systems



http://git-scm.com/book/en/Getting-Started-About-Version-Control

Linus Torvalds

http://www.ubuntu-user.com/Online/News/Linus-Torvalds-Windows-7-Rocks

Linus Torvalds talking with **Aalto** University students

It’s a free and open source **d**istributed **v**ersion **c**ontrol **s**ystem designed to handle everything from **small** to very **large projects** with **speed** and **efficiency**.

http://git-scm.com/

About the name **git**

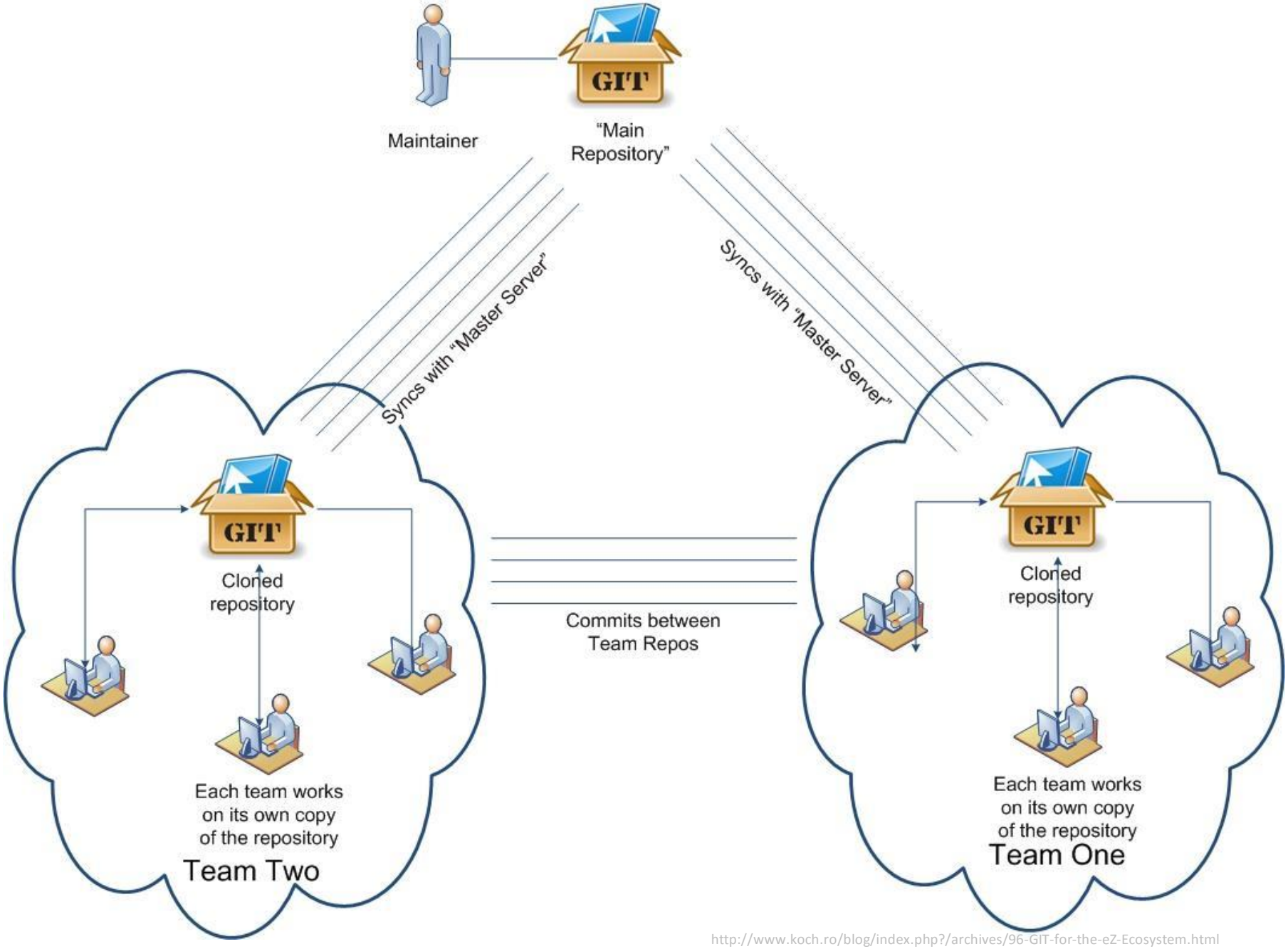
In British English slang roughly equivalent to „**unpleasant person**”.

Torvalds said:

„I'm an egotistical bastard, and I name all my projects after myself.

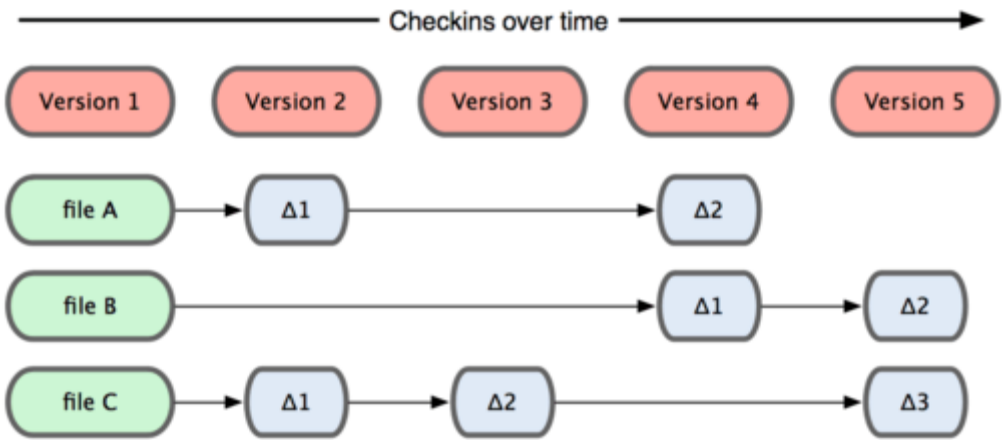
*First **'Linux'**, now **'git'**.”*

http://en.wikipedia.org/wiki/Git_(software)



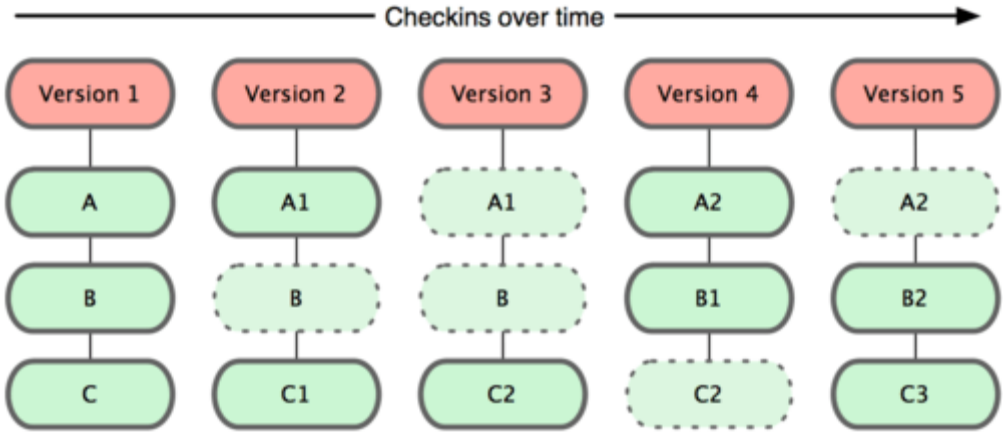
http://www.koch.ro/blog/index.php?/archives/96-GIT-for-the-eZ-Ecosystem.html

Store data as changes to a base version of each file



http://git-scm.com/book/en/Getting-Started-Git-Basics

Git stores data as snapshots of the project over time.



http://git-scm.com/book/en/Getting-Started-Git-Basics

Who uses git?

- Linux Kernel
- YUI3 (YAHOO!)
- android (Google)
- X.org and related projects, CompizGNU Autoconf, Automake, core utils
- prototype (Javascript library)
- Ruby on Rails
- Qt (Trolltech/Nokia)
- …

Getting and creating projects

- `git` **`init`**
- `git` **`clone`**

`git` **`init`**

Create an empty git repository or reinitialize an existing one.

`git` **`clone`** `<path>`

Clone a repository into a new directory.

Basic snapshotting

- `git` **`add`**
- `git` **`status`**
- `git` **`diff`**
- `git` **`commit`**
- `git` **`reset`**
- `git` **`rm`**
- `git` **`mv`**

git **add** .

```
git add -A (--all)
    git add -u
git add <filename>
```

Add file contents to the index

`git` **`status`** `<path>`

Show the working tree status.

`git` **`commit`** `-m „info”`

Show the working tree status.

It’s equivalent

```
git commit -a -m „info”
```

```
git commit -am „info”
```

```
git add -u
git commit -m „info”
```

git is forgiving

```
git commit -a -m „info”
```

Oops, we want to change the contents of the commit.

git is forgiving

```
git commit -a -m „info”
```

Oops, we want to change the contents of the commit. No problem ☺.

```
git add newFile
git commit --amend -m „new info”
```

Branching and merging

- `git` **`branch`**
- `git` **`checkout`**
- `git` **`merge`**
- `git` **`mergetool`**
- `git` **`log`**
- `git` **`stash`**
- `git` **`tag`**

`git` **`branch`**

List, create, or delete branches.

`git` **`checkout`**

Checkout a branch or paths to the working tree.

It’s equivalent

```
git branch new_branch
git checkout new_branch
```

```
git checkout -b new_branch
```

`git` **`merge`**

Join two or more development histories together.

`git` **`log`**

Show commit logs.

`git` **`stash`**

Stash the changes in a dirty working directory away.

`git` **`tag`**

Create, list, delete or verify a tag object signed with GPG.

Sharing and updating projects

- `git` **`fetch`**
- `git` **`pull`**
- `git` **`push`**
- `git` **`remote`**
- `git` **`submodule`**

`git` **`fetch`**

Download objects and refs from another repository.

`git` **`pull`**

Fetch from and integrate with another repository or a local branch.

`git` **`push`**

Update remote refs along with associated objects.

`git` **`remote`**

Manage set of tracked repositories.

Best practices

- **`commit`** often
- **`pull`** often
- use **`checkout`** and **`reset`** with caution
- create your own repository anywhere

http://hoth.entp.com/output/git_for_designers.html

git-tfs

It is a two-way bridge between TFS and git, similar to git-svn.

Git vs TFS

Git	TFS
Commit + Push	Check-In
Pull	Get Latest Version
Clone	‘Map Local Path’
Stash (only local though)	Shelve
Tag	Label
Fetch	‘Compare Local to Server’

http://johanleino.wordpress.com/2013/09/18/tfs-vs-git-or-is-it-tfs-with-git/

Try git

Code School - Try Git

try.github.io/levels/1/challenges/1

1.1 · Got 15 minutes and want to learn Git?

Git allows groups of people to work on the same documents (often code) at the same time, and without stepping on each other's toes. It's a distributed version control system.

Our terminal prompt below is currently in an **octobox** directory. To initialize a Git repository here, type the following command:

```
git init
```

tryGit

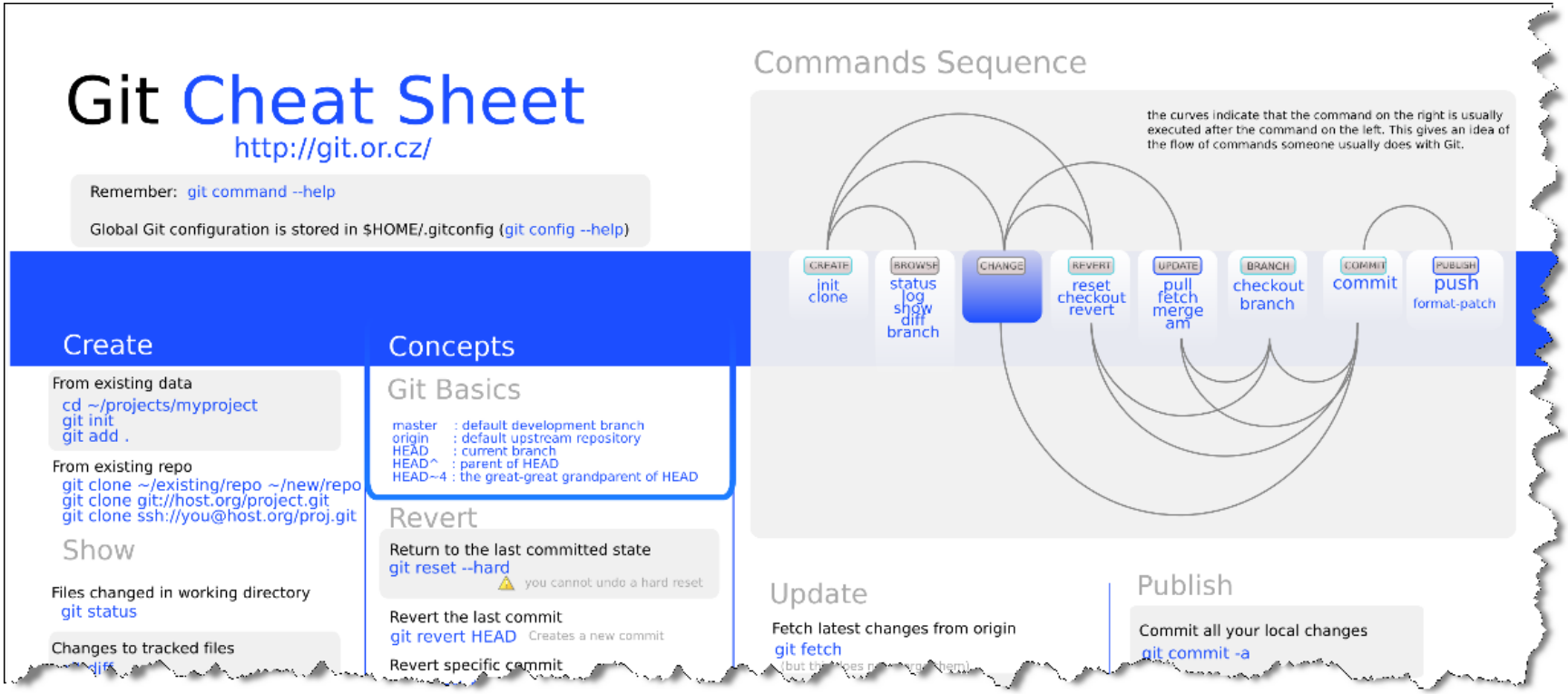
TryGit—832x310

```
Press enter to submit commands

>
```

http://try.github.io

Git Cheat Sheet



http://ktown.kde.org/~zrusin/git/git-cheat-sheet-large.png

In summary

- Git is fast
- Git is distributed
- Git is flexible

Resources

- **http://git-scm.com/documentation**
- **Pro Git** by *Scott Chacon.*
- **Pragmatic Guide to Git (Pragmatic Programmers)** by *Travis Swicegood.*
- **Pragmatic Version Control Using Git (Pragmatic Starter Kit)** by *Travis Swicegood*.