



git

Git Version Control

Conf. univ. dr. Catalin Boja

catalin.boja@ie.ase.ro

<http://acs.ase.ro>



Dep. de Informatică și Cibernetică Economică
ASE București

De ce controlul versiunilor ?

- Proiectele software se dezvoltă în regim colaborativ, codul fiind controlat de mai mulți programatori
- Ai nevoie de o soluție de back-up independent de mașina pe dezvoltare folosită
- Ai nevoie de o platformă care să permită lucru colaborativ (fiecare lucrează pe mașina locală și apoi modificările sunt încărcate în sistem pentru a fi disponibile tuturor membrilor echipei)
- Ai nevoie de o platformă care să îți permită accesul la istoricul modificărilor
- Toată lumea folosește o soluție de control al versiunii (soluție de versionare)
- *Este un instrument super cool*



“Only wimps use tape backup. REAL men just upload their important stuff on ftp and let the rest of the world mirror it.”

Linus Torvalds

Istoric

- Dezvoltat de Linus Torvalds pentru a gestiona proiectul de dezvoltare a kernel-ului de Linux ~2005
- Git (n) – *British slang for a stupid or unpleasant person*
- Adoptat rapid de companii datorita avantajelor oferite

Linked  [®]

Google

facebook [®]

Microsoft

Concepte

VCS – Version Control Software

SCM – Software Control Management

SCM – Source Code Management

Repository (Repo) – baza de date/imagini/colecție de fișiere din proiect

Branch – Imagine locală a proiectului. Permite lucrul pe versiuni paralele fără a afecta imaginea principală a proiectului

Stash – arhivă locală pentru un set de modificări

Commit – salvare modificări pe server

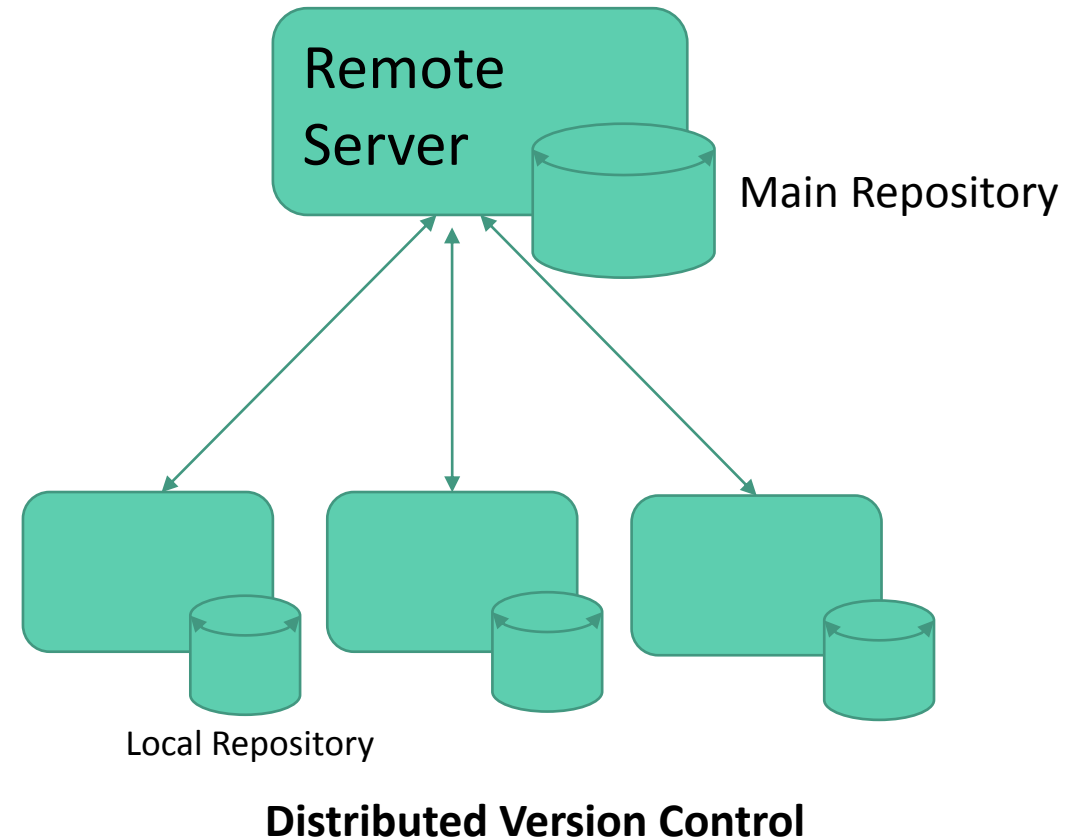
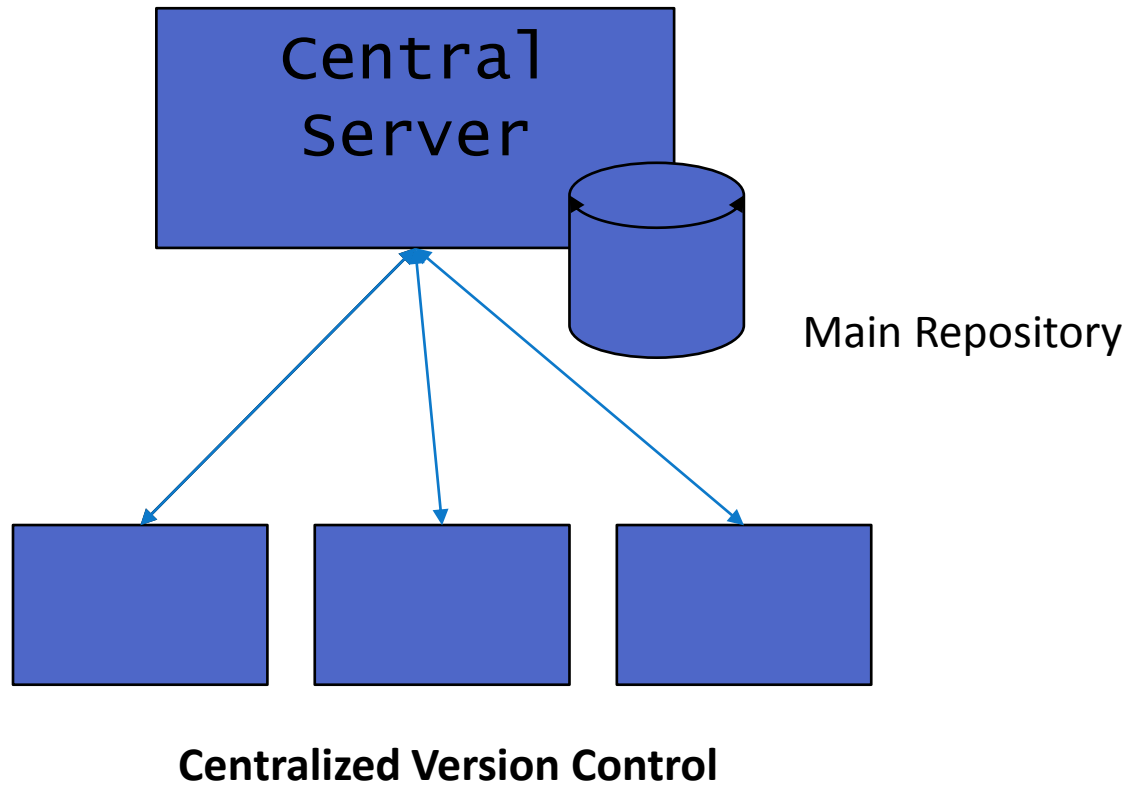
Ce este Git ?

- An open source version control system (VCS)
- A distributed version control system
- A directory content management system
- A tree based history storage system
- A *Stupid content tracker* (Linux man page - <http://linux.die.net/man/1/git>)

Caracteristici Git

- Este un sistem distribuit
- Toți au acces la istoricul modificărilor
- Toți lucrează pe mașinile de dezvoltare in mod offline
- Nu exista o autoritate centrala de control – toți au aceleași drepturi
- Modificarile pot fi transmise si in mod P2P (peer-to-peer) fără a trece prin server
- Este un sistem distribuit de control al versiunii (Distributed Version Control – DVC)
- Fișierele sunt urmărite prin verificarea hash-ului (SHA-1) acestora

CVC vs DVC



CVC vs DVC

CVC and Subversion:

- SVN
- CVS
- Alte soluții comerciale

DVC:

- Mercurial
- Git
- BitKeeper
- Darcs

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

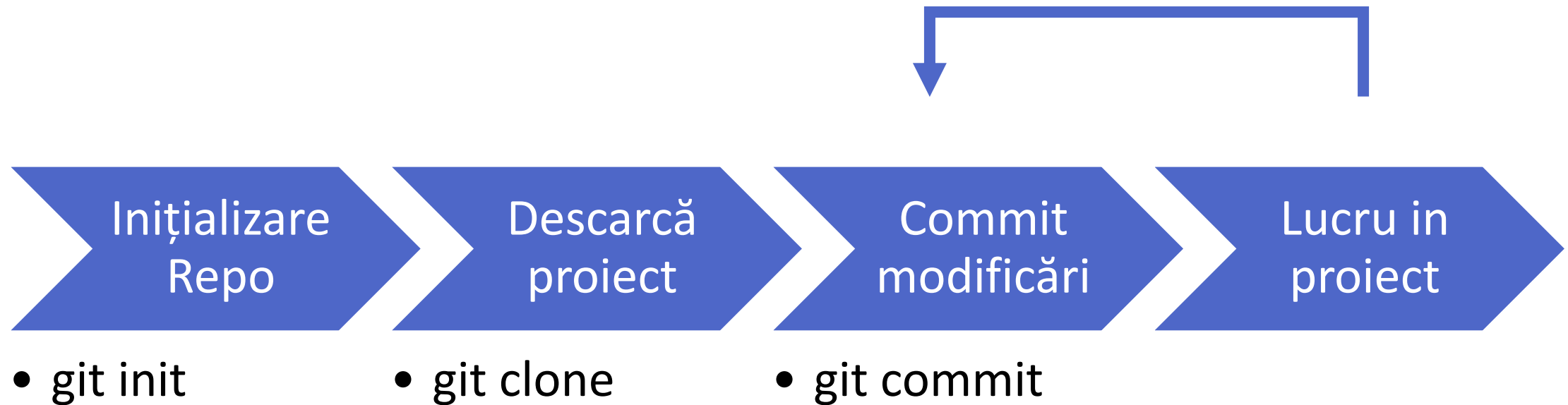
Resurse Git

- Instrumente: <http://git-scm.com/downloads>
- Tutoriale:
 - Cautare dupa “Git tutorial” pe Google
 - <https://try.github.io/levels/1/challenges/1>
 - Documentatie oficiala: <http://git-scm.com/doc>
 - GitHub Cheat Sheet: <https://training.github.com/kit/downloads/github-git-cheat-sheet.pdf>
 - Cartea *Pro Git editia 2*: <http://git-scm.com/book/en/v2>

Operatii de baza Git (VCS)

- Inițializare repository
- Adaugă fișiere noi/modifica fișiere existente
- *Commit* – salvează modificările pe server
- Istoric - vizualizare istoric modificări
- Share – distribuie modificările
- Update – actualizează proiectul cu ultimele modificari
- Revert - anulează modificări cu revenire la o versiune anterioara
- Branch – creare ramura (proiect secundar) de dezvoltare locala

Flux Git



Tutorial Git

1. Creare cont si proiect pe <https://github.com> sau pe <https://bitbucket.org/>
2. Creare proiect: <https://github.com/catalinboja/cts-sourcecode>
3. Instalare client Git (Git Bash) pe masina de lucru
4. Creare director repo local
5. Deschide client Git pe directorul creat
6. Inițializare Git cu `git init`
7. Clonare repo server cu `git clone https://github.com/catalinboja/cts-sourcecode.git`

Tutorial Git

1. Creează repo local

- `>mkdir CTSRepo`
- `>cd CTSRepo`

2. Inițializează client Git

- Deschide clientul Git (Bash in linie de comanda) in directorul *CTSRepo*
- `git init`

Tutorial Git

3. Configurare client

- `git config --global user.name "Your Name Here,"`
- `git config --global user.email your_email@youremail.com`

4. Definiște repo-ul master

- `git remote add origin https://catalinboja@bitbucket.org/catalinboja/cts_courses_examples.git`

Tutorial Git

5. Descărcare fișiere din repo-ul master (se creează o copie locala)

- `git clone https://github.com/catalinboja/cts-sourcecode.git`

6. Definiște un branch local si lucrează pe el

- `git branch work`
- `git checkout work`
- `git rebase master`

7. Modifică fișiere locale si încarcă-le pe server

- `git status`
- `git add .`
- `git commit -a -m "Modificari fisiere"`

Tutorial Git

8. Treci pe branch-ul *master* si încarcă modificările

- `git checkout master`
- `git pull`
- `git rebase work`
- `git push`

9. Treci pe branch-ul *work*

- `git checkout work`

simple daily git workflow



<https://www.sonassi.com/knowledge-base/our-magento-git-guide-and-work-flow/>