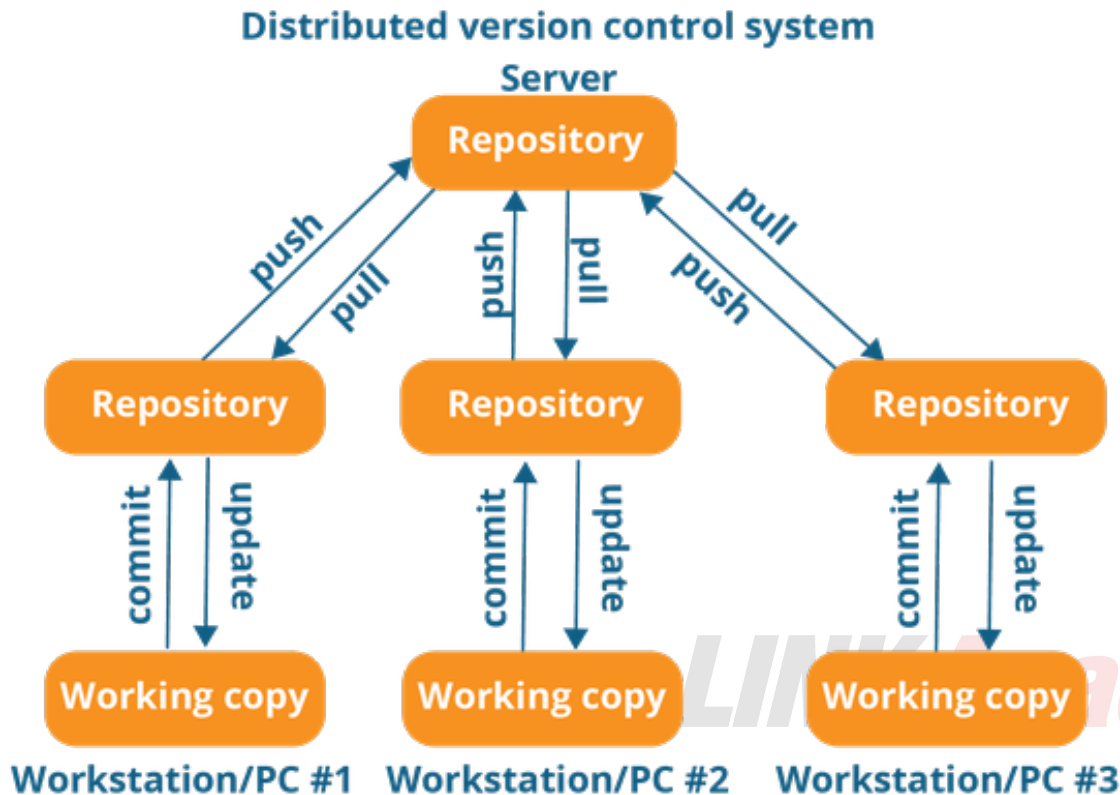


Sistem de versionare



Sistem de versionare

- ✓ Sisteme de versionare: permit gestionarea versiunilor multiple a unor fisiere, precum si asigurarea lucrului colaborativ asupra acestor fisiere.
- ✓ Fiecare modificare va fi inregistrata continand atat diferentele noii versiuni (revision), precum si autorul acestei noi versiuni
- ✓ Au fost concepute pentru a permite membrilor mai multor echipe sa opereze modificari pe acelasi proiect, aceste modificari urmand a fi reunite intr-o noua versiune a proiectului

Sistem de versionare

- ✓ Source Code Control System (SCCS)
- ✓ Revision Control System (RCS)
- ✓ Concurrent Versions System (CVS)
- ✓ Subversion (SVN)
- ✓ [Git](#)
- ✓ Mercurial

Sistem de versionare

✓ Terminologie

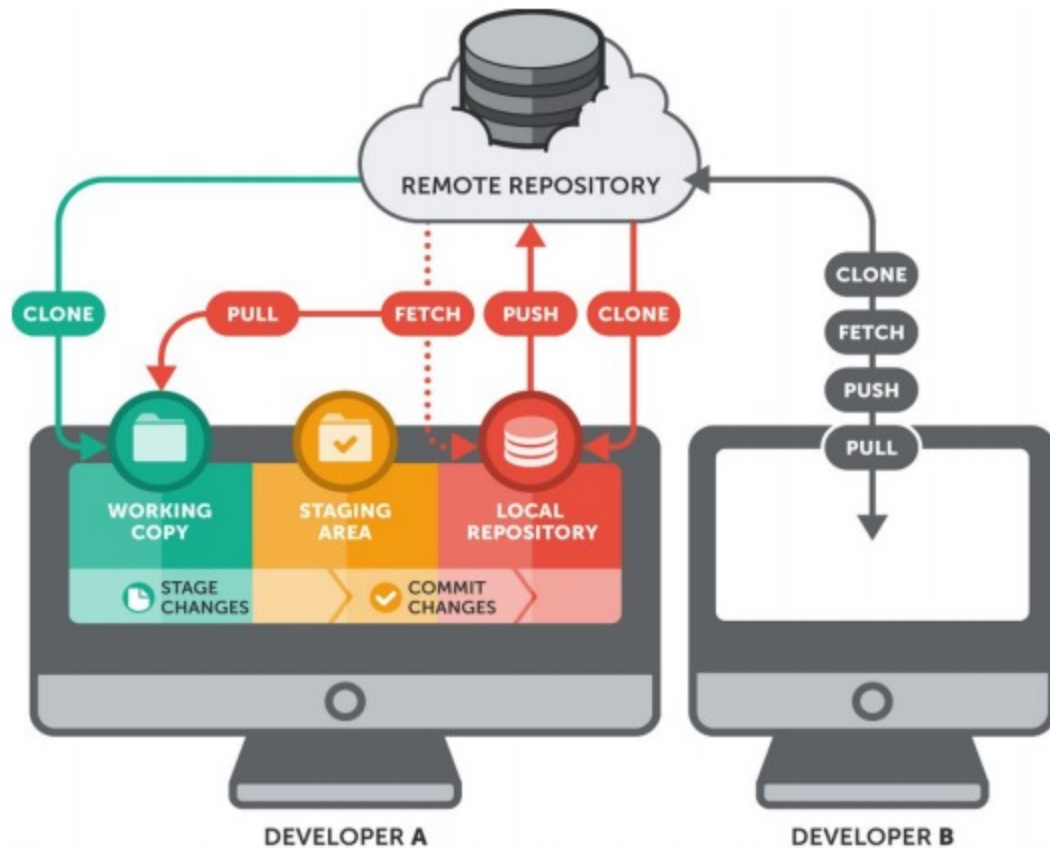
- **repository**: componenta server ce contine informatii privind ierarhia de fisiere si reviziile
- **checkout**: preluarea in mediul local a unei anumite revizii publicate pe server
- **working copy**: versiunea locala a proiectului
- **commit**: cerere de publicare pe server a unor modificari
- **Pull / push** : actiunea de actualizare (update) a informatiilor locale cu cele de pe server / si invers
- **conflict**: apare atunci cand mai multi utilizatori vor sa publice modificari aplicate acelorasi fisiere din proiect, insa sistemul de aplicare a versiunilor diferite nu poate imbina modificarile
- **revert**: revenirea la o versiune anterioara pe un anume fir de dezvoltare (branch)
- **branch**: ramuri secundare de dezvoltare a proiectului

Sistem de versionare

- ✓ **GIT**: este un sistem de versionare distribuit ce ruleaza, sub licenta GNU

<https://git-scm.com/>

- ✓ Autorul lui este Linus Torvalds, parintele Linux, cel care l-a conceput in 2005 ca urmare a conflictului aparut cu BitKeeper, vechiul sistem de versionare folosit pentru kernelul de Linux.



Sistem de versionare

✓ Cine foloseste GIT :

Linked  in.

Google

facebook.

Microsoft

LINK Academy

Sistem de versionare

✓ Caracteristici Git:

- ✓ Este un sistem distribuit
- ✓ Toti au acces la istoricul modificarilor
- ✓ Toti lucreaza pe masinile de dezvoltare in mod offline
- ✓ Nu exista o autoritate centrala de control – toti au aceleasi drepturi
- ✓ Modificarile pot fi transmise si in mod P2P (peer-to-peer) fara a trece prin server
- ✓ Este un sistem distribuit de control al versiunii (Distributed Version Control – DVC)
- ✓ Fisierile sunt urmarite prin verificarea hash-ului (SHA-1) acestora

✓ Resurse Git:

<https://help.github.com>

<https://try.github.io/levels/1/challenges/1>

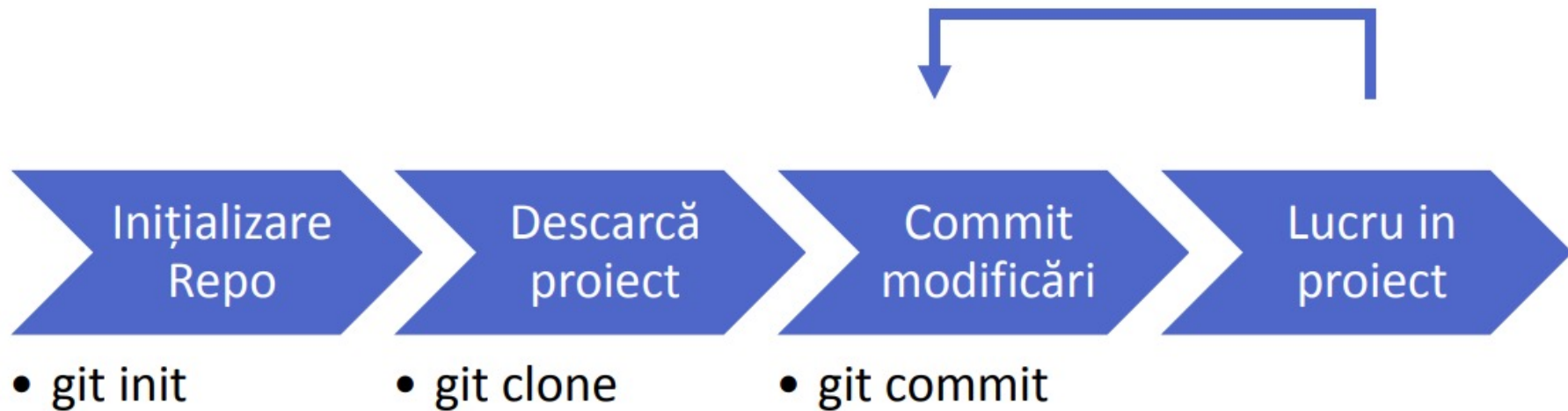
<http://git-scm.com/doc>

<https://training.github.com/kit/downloads/github-gitcheat-sheet.pdf>

<http://git-scm.com/book/en/v2>

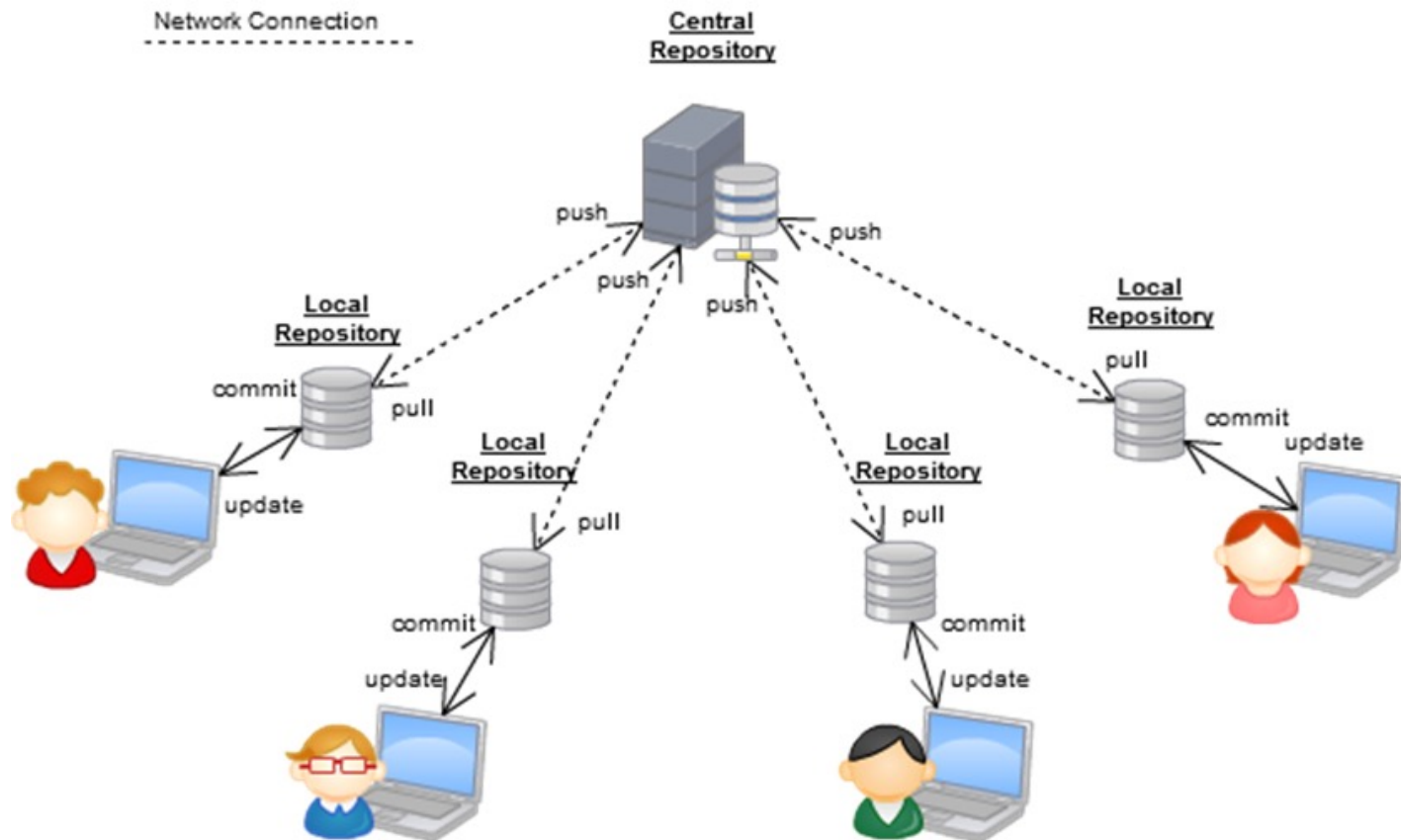
Sistem de versionare

✓ Flux Git



Sistem de versionare

✓ Flux Git



Sistem de versionare

1. Creare cont si proiect pe <https://github.com> sau pe <https://bitbucket.org/>
2. Creare proiect: <https://github.com/USERGITHUB/NUMErepo>
3. Instalare client Git (Git Bash) pe masina de lucru
4. Creare director repo local
5. Deschide client Git pe directorul creat
6. Initializare Git cu **git init** (**.git**)
7. Clonare repo server cu

git clone <https://github.com/USERGITHUB/NUMErepo.git>

Sistem de versionare

1. Creeaza repo local

>**mkdir test** (apoi vom testa cu test.com)

>**cd test**

2. Initializeza client Git

Deschide clientul Git (Bash in linie de comanda) in directorul test

git init

Sistem de versionare

3. Configurare client

git config --global user.name "Numele vostru"

git config --global user.email ["email@youremail.com"](mailto:email@youremail.com)

4. Defineste repo-ul master

git remote add origin <https://github.com/USERGITHUB/NUMErepo.git>

git remote set-url origin

<https://github.com/USERGITHUB/NUMErepo.git>

Verificare: **git remote -v**

Sistem de versionare

5. Descarcare fisiere din repo-ul master (se creeaza o copie locala)

git clone <https://github.com/USERGITHUB/NUMErepo.git>

lucreaza pe el

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

7. Modifica fisiere locale si incarcă-le pe server

git status

git add . `git add -u` `git add -A`

git commit -m "Modificari fisiere"

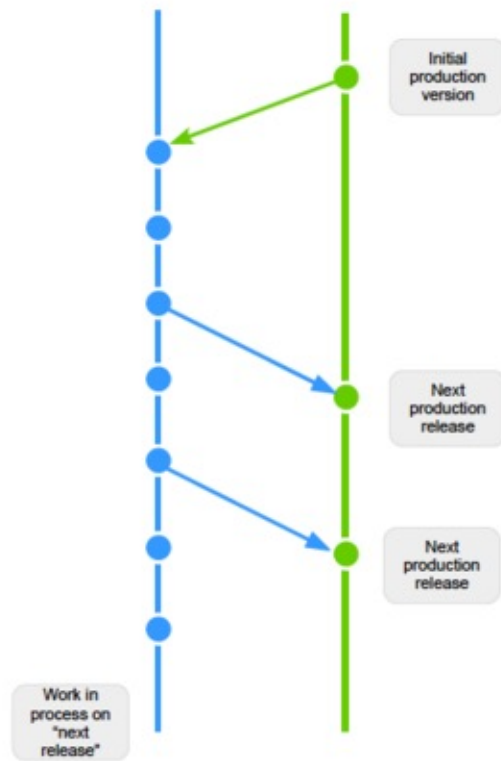
git push origin main

8. Treci pe branch-ul master si incarca modificarile

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

9. Treci pe branch-ul work `git checkout work`

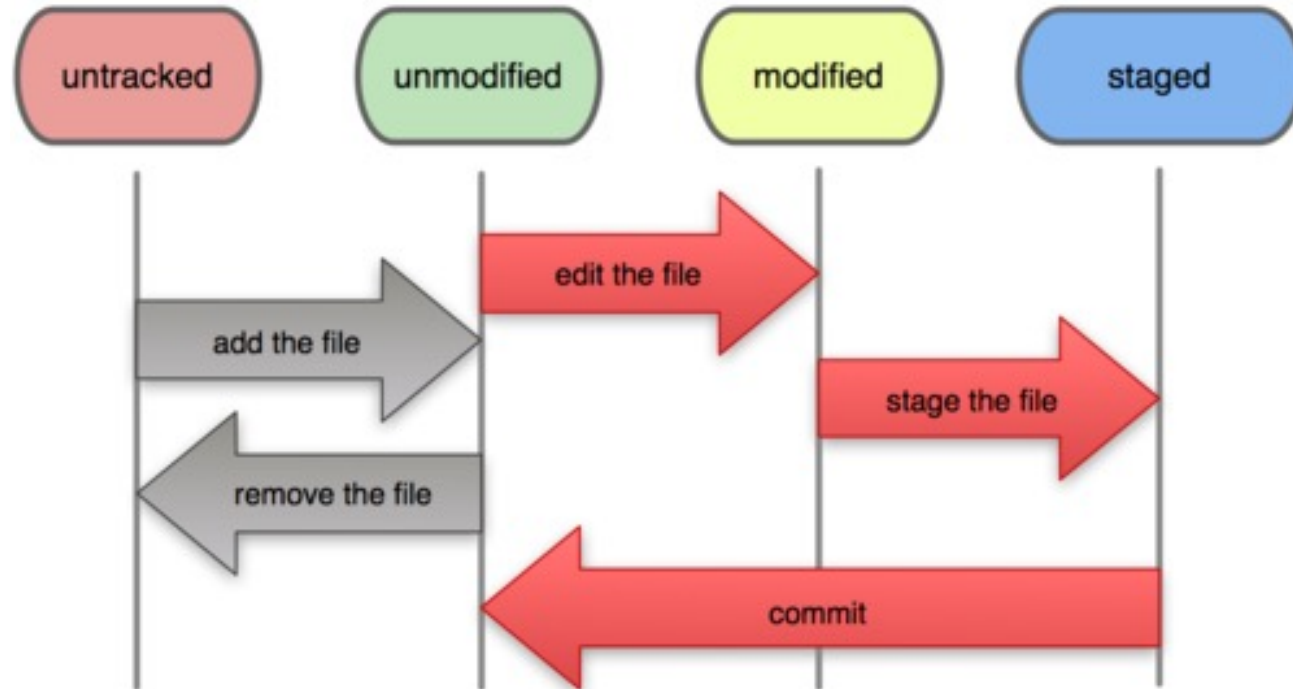
Main branches
development master



LINK

Sistem de versionare

File Status Lifecycle



Sistem de versionare

✓ Starea fișierelor

- git status
- git log
- git diff

✓ Fișierele sau directoarele ce se doresc a fi ignorate de sistemul de versionare pot fi trecute într-un fișier numit .gitignore.

<https://help.github.com/articles/set-up-git/>

<https://help.github.com/articles/adding-an-existing-project-to-github-using-the-command-line/>

<https://help.github.com/articles/inviting-collaborators-to-a-personal-repository/>