

Ministerul Educației și Cercetării al Republicii Moldova
IP Colegiul “Iulia Hasdeu” din Cahul

Asistență pentru managementul proiectelor software

Tema 5: Refacerea istoricului

Crearea etichetelor - git tag

Apareci Aurica
Babaianu Savelie
Grupa: AAW 2042

The background is a dark navy blue. On the left, there is a faint, light blue Git logo consisting of a diamond shape with a branching diagram inside. To the right of the logo, the word "git" is written in a large, lowercase, light blue sans-serif font, serving as a watermark. Centered on the left side of the image is the text "Git history" in a bold, white, sans-serif font, with the word "rewrite" in a smaller, bold, white, sans-serif font directly below it.

Git history

rewrite

Refacerea istoricului

Refacerea istoricului (Git History Rewrite) se referă la procesul de modificare a istoricului commit-urilor într-un depozit Git. Acest proces poate include schimbarea sau eliminarea commiturilor existente, restrângerea sau extinderea istoricului, și alte manipulări ale istoricului commit-urilor pentru a satisface cerințele sau pentru a face istoricul mai curat și mai organizat.


1 **Modificări suplimentare la commit-ul anterior**

`git commit --amend`

2 **Reorganizarea istoricului commit-urilor**

`git rebase bază`

v



Comanda **`git commit --amend`** este folosită pentru a face modificări suplimentare sau pentru a adăuga schimbări la commit-ul anterior. Aceasta este utilă atunci când dorești să adaugi sau să corectezi ceva în cel mai recent commit fără a crea un commit nou.

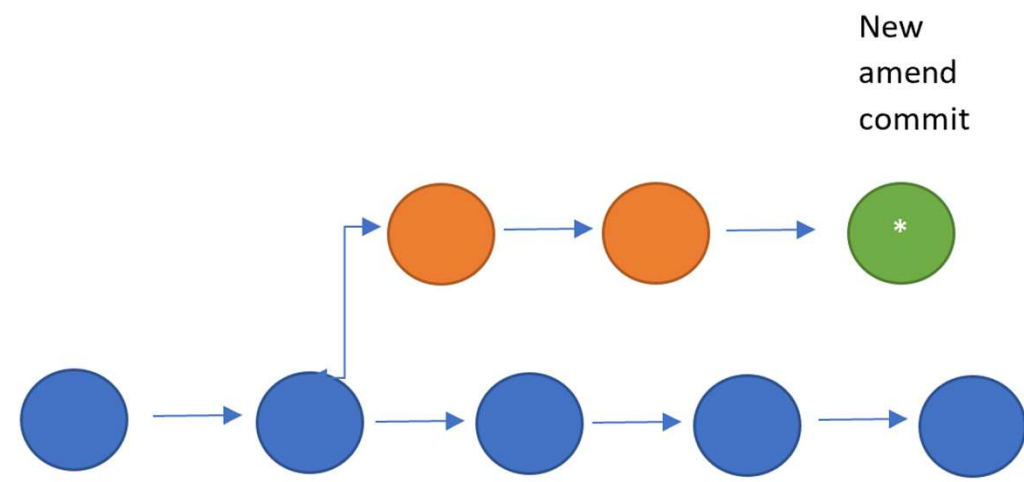
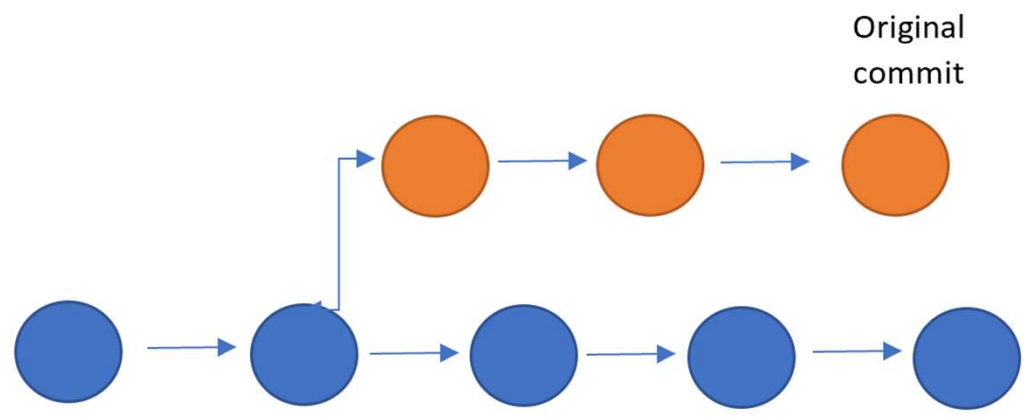
Fiecare commit este listat cu un prefix care indică acțiunile pe care le poți efectua:


pick: poate fi folosit pentru a păstra commit-ul neschimbat.

reword: poți modifica mesajul de commit. După selectarea acestei opțiuni, vei putea să editezi mesajul de commit în modul interactiv.

edit: poți modifica commit-ul. După selectarea acestei opțiuni, rebase-ul se va opri înainte de commit-ul respectiv, permițându-ți să faci modificări la fișierele din acel commit.

squash sau fixup: aceste opțiuni permit combinarea unui commit cu commit-ul precedent.

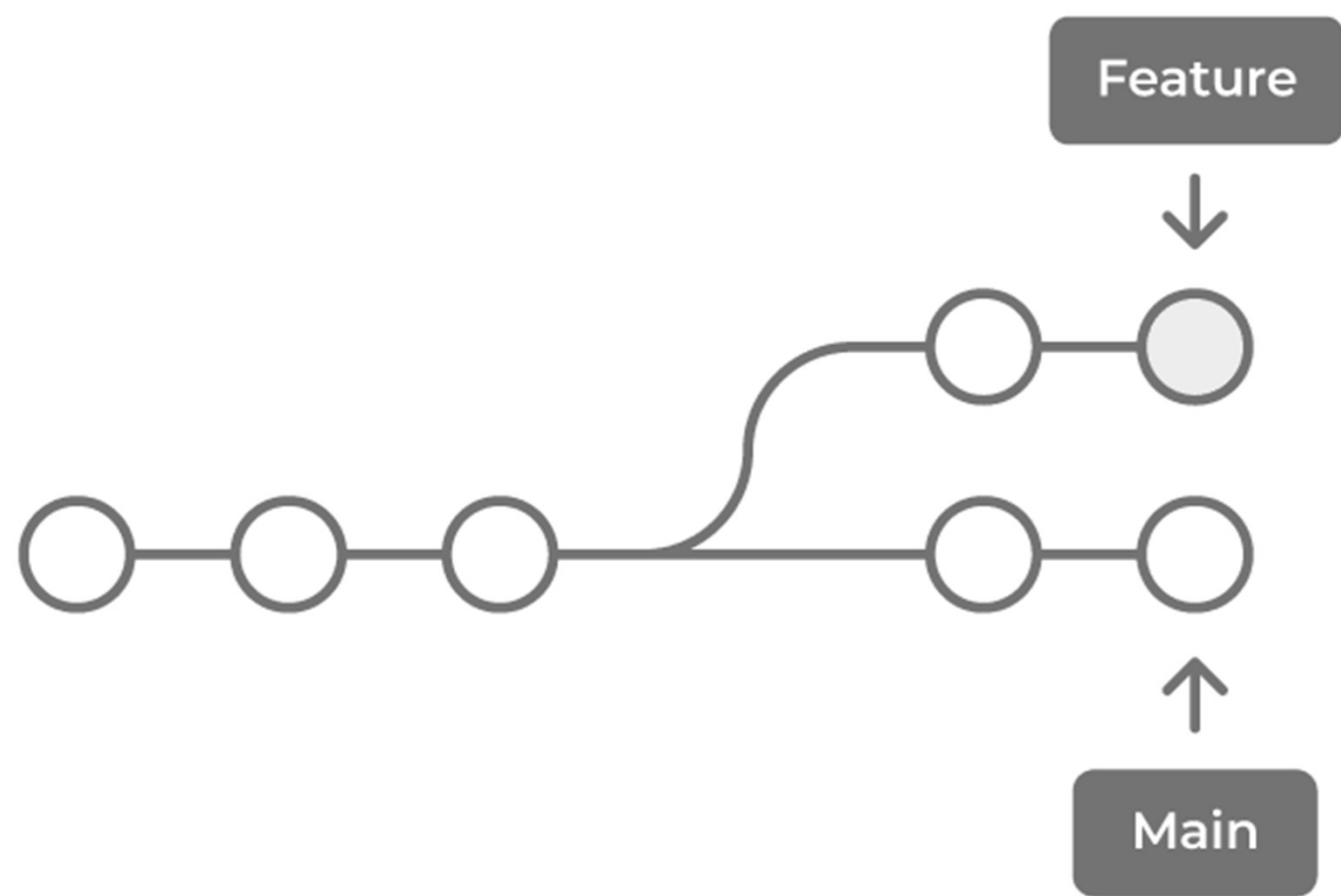




Comanda **`git rebase`** este folosită în Git pentru a restrânge sau reorganiza istoricul commit-urilor. Acest proces implică mutarea sau regăzduirea commit-urilor pe o altă ramură sau pe o altă bază.

Există două scenarii principale în care **`git rebase`** este adesea folosit:

1. Rebasing pentru a integra modificări dintr-un branch în altul: poți face rebase pentru a integra schimbările dintr-un branch într-un alt branch într-un mod liniștit și organizat.
2. Rebasing pentru a curăța istoricul commit-urilor: poți face rebase pentru a elimina commit-uri inutile sau pentru a combina commit-uri mici într-unul singur pentru a menține istoricul mai clar și mai ușor de citit.





Crearea și aplicarea etichetelor

GIT tag

Etichetele (tags) în Git sunt puncte de referință fixe în istoricul depozitului care sunt utilizate pentru a marca puncte semnificative în dezvoltarea proiectului. Acestea sunt utile pentru a identifica versiuni specifice, lansări sau momente cheie în istoricul commit-urilor.

Comenzile menționate sunt folosite pentru a crea, gestiona și aplica etichete în Git.

- 1 **git tag <nume_etichetă>** permite crearea unei etichete simple cu un nume dat. Aceasta marchează punctul curent în istoricul depozitului cu numele specificat.
- 2 **git tag -a <nume_etichetă> -m "mesaj_etichetă"** creează o etichetă cu un mesaj asociat, care poate oferi detalii despre lansare sau versiunea marcată.
- 3 **git show <nume_etichetă>** afișează informații despre o etichetă specifică, inclusiv commit-ul la care este ancorată.
- 4 **git tag -l** afișează o listă a tuturor etichetelor din depozit.

