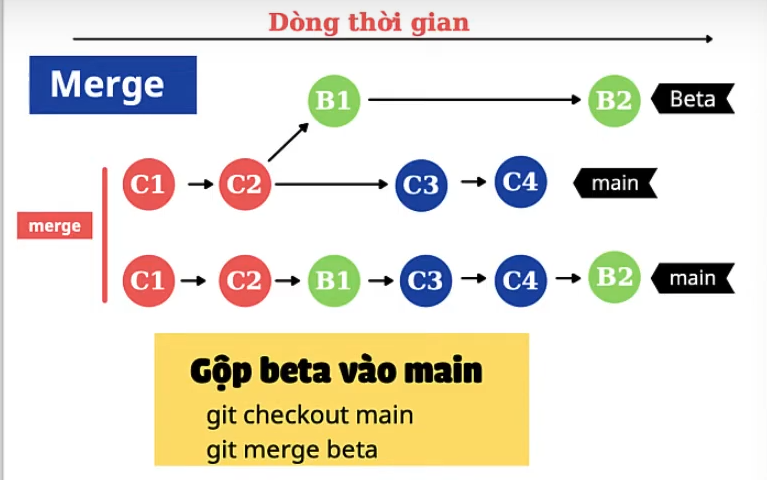
**TẤT TẦN TẬT VỀ GIT**

**1. Khai báo ban đầu**

git config --global user.name <username>

git config --global user.email <mailaddress>

**2. Git Merge**



Nó sẽ merge từ nhánh Beta vào nhánh Main theo thứ tự thời gian commit – tức là nếu trên nhánh Beta có commit mới mà nhánh main nó có thay đổi (C3 – C4) thì khi merge từ beta vào main thì B1 sẽ đứng sau C3 – C4.

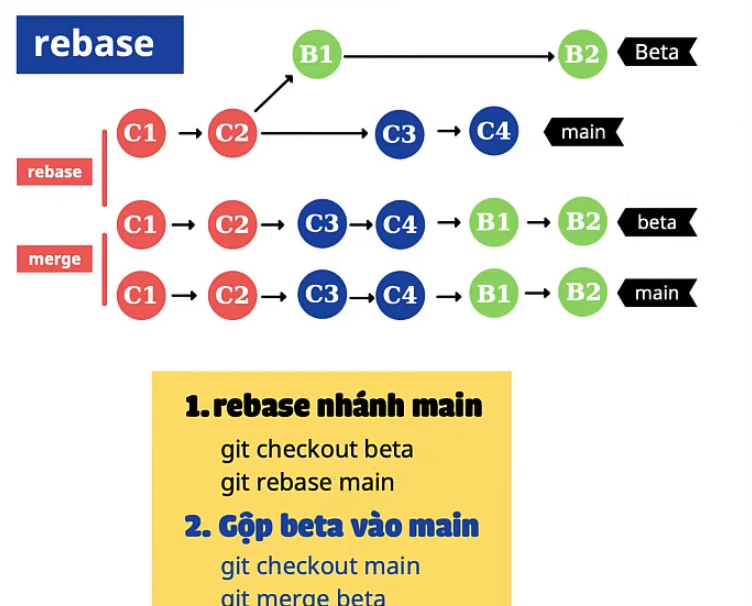
Conflic: Xảy ra khi 2 người cùng thêm code trên cùng 1 file.

* HEAD: Thay đổi ở file ở nhánh hiện đang đứng
* Icome Change: Dòng mới của nhánh mới.

=> Cần resolve: Muốn giữ cả 2 thì tiến hành (***git add .*** */* ***git push***)

Nếu k có conflic tiến hành push luôn.

**3. Git Rebase**



Rebase nhánh Main: giữ lại commit của nhánh main => đẩy 2 commit mới của nhánh Beta lên đầu.

Conflic: Xảy ra khi 2 người cùng thêm code trên cùng 1 file.

* HEAD: Thay đổi ở file ở nhánh hiện đang đứng
* Income Change: Dòng mới của nhánh mới.

\*Nhớ kiểm tra thay đổi: git status

=> Fix: **git add .** / **git rebase –continue**

Kiểm tra thứ tự commit: **Git log –oneline** (mới ở local , chưa lên git)

* **Git push –force**

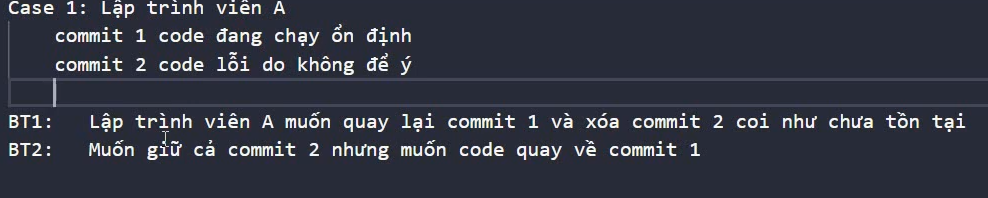
=> muốn đồng bộ nhánh main(hiện tại có 4 commit) với beta(đã add thêm 2 commit mới lên đầu) thì checkout tới nhánh main sau đó merge lại nhánh beta.

**4. Git Revert/Reset: Commit bị lỗi – quay về commit cũ**

***Reset***:

* **git reset --hard <tên commit>** *(khá nguy hiểm: nó xóa lịch xử đi)*
* **git push -f** *(ghi đè lên Repo) (hiểu mình đang làm gì thì dùng)*
* **Xóa luôn, không còn lịch sử nữa.**

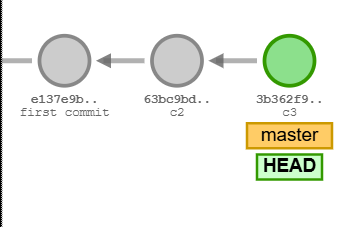
Checkcommit: git log/ git log -oneline



BT1: -- git reset (bỏ đi comit cũ)

BT2: -- git revert (lấy code cũ đẩy lên trc)

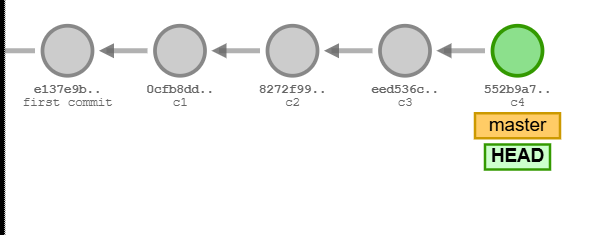
EX:



Xóa thằng commit c3 đi: ***git reset –hard <tên commit>***

**Revert: Lấy bản sao của commit trước đó đẩy lên ( An toàn hơn reset )**

* **git revert HEAD/id commit ( id, head ở đâu thì lấy commit trc nó, muốn lấy id commit 2 nó sẽ tạo một commit mới bản sao của commit 1)**



Tình huống: C3 ngon nghẻ rồi, C4 bị lỗi => muốn quay về C3 nhưng k xóa đi C4

Ở đây bản sao là c3 đó, revert thì nó sẽ đẩy bản sao của c3 lên trước c4 là c3’(thực chất là c5).

**After revert:**

