**Checkout - Note**

1. Checkout đến branch

Ví dụ: 1 2 3(master) 4 5(HEAD🡪todo-app-2.0)

6 7(todo-app-1.1)

* git checkout todo-app-1.1

Kết quả: HEAD 🡪 todo-app-1.1

Quá trình:

Trước khi checkout nên thực hiện git status. Nếu xuất hiện current tree clean có nghĩa là tất cả các file để ở trạng thái unmodified khi đó git sẽ cho phép checkout dễ dàng

Nếu tồn tại một vài file ở trạng thái untracked, modified, staged thì khả năng(tùy trường hợp) git sẽ không cho mình checkout và báo lỗi

Giải pháp:

Làm cho current working directory clean bằng 1 trong những cách sau

* git add, git commit để lưu lại nội dung thay đổi
* git stash … để lưu lại tạm thời chưa muốn commit
* git restore … để bỏ qua những phần đã thay đổi, sử dụng mình không muốn giữ lại phần thay đổi

2. Checkout đến commit hash

Ví dụ: 1 2 3(master) 4 5(HEAD🡪todo-app-2.0)

6 7(todo-app-1.1)

* git checkout 6

Kết quả: HEAD 🡪 6 – trạng thái detach head

Ở trạng thái này khi mình thay đổi code và commit giả sử commit có mã hash 8 sẽ đi ra từ 6 và HEAD trỏ đến 8. Không có bất kỳ branch nào trỏ để commit 8. Sau đó nhành HEAD về 1 nhánh bất kỳ. Gõ git log –online –all thì mình cũng sẽ không tìm thấy commit 8. Dễ mất code ở commit 8. Muốn lấy lại vẫn được dùng git reflog nhưng hơi mất công và ít khi thực hiện. Cho nên khi nào thực hiện cũng nên đưa HEAD về branch nào đó xong tiếp tục code và commit để branch nhảy lên. Thông qua branch mình sẽ biết đang là version nà của dự án và version đó có những commit, chức năng nào