At home

First of all, you fetch data (the git repository) from the Image-data to your computer (step ①), and then you can do any other edit. When you end of the homework, you push the local git-repo to the Home-data that on the Core data server (step ②).

At company

If you did homework at home, then you will fetch the work-result from Home-data to you computer (local PC)(step ③), else fetch the repo-data from company server (source-data)(step ④ or ⑥).

When you end something edit, you push the result to the company server (step ⑤ or ⑦).

Synchronize

The company server will push the repo-data to the core data automatically (step ⑧). And the core data is read only for any other developers.

It shows that even you at home, you can continue your working, that’s independent on the company’s net. More about independent on the net will be descript later.

②

①

③

⑧

④

⑤

⑥

⑦

Working architecture 1

Company w-f

DevＡ

DevＢ

Core data

某个代理服务器

Home

Home dev

How does git working?

Working architecture 2

Work area is our edit area (the current project in the IDE).

Version repo is the stage area (sometimes it is not saved in the objects).

Master is the storage area (that point to the Objects, that already saved in the files). And the HEAD is a pointer that point to the master.

1) Is synchronizing work area to stage, this will using work’s content cover the stage’s content. But not affect to the master (or any other branch that point to objects), to implement this using this command “git add ….”.

2) This is counter example with 1). Not affect to the master also. Implement command “git checkout”.

3) Clear the stage’s content, this not affect neither work area nor master.

4) Synchronizing stage to the master, in other word copy the stage to master.

5) Copy the head current point branch to stage. Implement command “git reset HEAD”.

6) Copy the head current point branch to stage and work area. Implement command “git checkout HEAD”.

7) The HEAD point to current activity branch automatically.

Work area

Version repo

Master

Objects

HEAD

7

6

5

4

3

2

1

A

B

C

Diff

A, Work area with stage. Command “git diff”.

B, Stage with HEAD. Command “git diff --cached”.

C, Work area with HEAD. Command “git diff HEAD”.