

Git

devops所需要知道的一些测试:

A/B Test

为什么要用version control?

撤销改动或者回滚版本snapshot

回溯历史 A complete long-term history of every file that provides traceability

协同合作 via branching strategies

Version Control Glossary:

Repository: your project. (repo)代码仓库 code + history

Diff: 两个版本之间的差异 两个snapshot之间的差异

Commit: A snapshot of your project's state at a point in history.commit之后的历史是不可以修改的。

Branch: Modifications to a project made in parallel with the main branch, but not affecting the main branch.

Merge: 一个branch的代码 merge到另一个代码里进行代码的合并。

Clone: Downloading a local copy of a project, project = repo(code + history)

Fork: 类似于repo的基本做branch, 以repo为基础的

Git workflow

Two steps workflow

Git - Download

Git - GUI图形化

Git and Github

1 Git Setup

Git Global Setup:

git config -- global user.name "Lin Lu"

git config --global user.email "chaolin1984@gmail.com"

git config --global color.ui auto

git config --global merge.conflictstyle diff3

git config --global core.editor "code --wait"

If enter code didn't work you can follow below:

1> open VS code

- 2> press cmd+shift +p to pop open the command palette on a mac, and command + shift +p on Windows
- 3> Search for “code” to find “shell command : Install ‘code’ command in path”
- 4> Select it and you’re all done!

Git init

git init : set up a new git repo and begin tracking an existing directory.

拉取repository的方法有两种:

- 1> git init
- 2> git clone (remote repository only)

git status git当前的状态

git add . 把文件加到staging area里面

git rm – cached <file name> 把文件移除staging area

git commit -m “my first git commit”

Two steps : git add . git commit

git log : 查看history

Gitignoring file:

you can create a file listing patterns to match them named **.gitignore**

Undoing thing command:

git commit –amend

git rm test.txt

git clean

git stash

Commit Message 一定要有意义 !

Semantic branch name:

<type>/<ticket-number>-<title>

eg: feat/JR-101-create-header-for-home-page

Git undo changes:

git checkout : undo modified file

git clean : 删除 git clean -f

git revert 用一个新的commit对历史记录进行回滚 soft - staging

git reset 从历史记录中删除commit hard

2 两条线的历史 - branch & merge

Git with branching:

create a branch (git branch)

checkout a branch (git checkout)

merge a branch (git merge)

rebase a branch (git rebase)

Branching:

git branch 列出所有分支 list all branches

git branch -d 删除分支 delete branch

git checkout -b 创建分支 create new branch

Semantic branch name:

<type>/<ticket-number>-<title>

eg: feat/JR-101-create-header-for-home-page

Merge a branch (git merge)

merge the target branch into current branch

Merge conflicts!

two steps : fix conflicts and the commit the result

pull before commit

merge from master before commit

Merge vs Rebase

rebase : A clean, linear history free of unnecessary merge commits.

3 多条线的历史 - 远程协作

Connect to remote repo

git clone

git remote add <name> <url>

git remote rm <name>

git remote rename <old-name> <new-name>

Pull Request (PR)

当我们合并代码的时候, 防止事情发生, remote

origin -> clone from -> GitHub -> remote

pull request怎么做?

1. 先把远程的代码push到本地 git clone + url
2. 新建本地branch eg: feat101/add-new-function
3. 然后git add and git commit
4. pull request到remote repo (需要授权一下)
5. 之后填写pr的信息选择reviewer
6. 等待reviewer的回复
7. 谁提的pr谁要等待ok后自己merge到main branch