### 2019.07.13学习笔记(p.m)

**Git相关：**

1. **分支管理：**

**1）master分支应该是非常稳定的，也就是仅用来发布新版本，平时不能在上面干活；**

**2）干活都在dev分支上，也就是说，dev分支是不稳定的，到某个时候，比如1.0版本发布时，再把dev分支合并到master上，在master分支发布1.0版本；**

**3）你和你的小伙伴们每个人都在dev分支上干活，每个人都有自己的分支，时不时地往dev分支上合并就可以了。**

**4）所以，团队合作的分支看起来就像这样：**



**5）合并分支时，加上--no-ff参数就可以用普通模式合并，合并后的历史有分支，能看出来曾经做过合并，而fast forward模式合并（git merge <name>）就看不出来曾经做过合并：**

**$ git merge --no-ff -m "merge with no--ff" dev**

**6）使用git log加上参数查看历史分支：**

$ git log --graph --pretty=oneline --abbrev-commit

\* d9b26da (HEAD -> master) add bbbb

\* b519854 merge with no--ff

|\

| \* c439e63 (dev) add aaaa

|/

\* 0f8061e conflict fixed

|\

| \* 6bac58d AND simple

\* | e194fca & simple

|/

\* 76bbcb9 add a line

\* 7a73e75 (origin/master, origin/HEAD) add test.txt

\* 02374b6 Initial commit

1. **使用git log查看历史记录时，查看后在英文状态下按Q退出（不分大小写）**
2. **bug分支管理：**

**修复bug时，我们会通过创建新的bug分支进行修复，然后合并，最后删除；**

**当手头工作没有完成时，先把工作现场git stash一下，然后去修复bug，修复后，再git stash pop，回到工作现场，操作如下所示具体：**

**1）使用git stash命令保存现在所在分支状态：**

$ git stash

Saved working directory and index state WIP on dev: c439e63 add aaaa

**2）查看工作区状态：**

$ git status

On branch dev

nothing to commit, working tree clean

**工作区处于干净状态**

**3）确定在哪个分支上修复bug（假设在master分支上），则需切换到master分支上：**

$ git checkout master

Switched to branch 'master'

Your branch is ahead of 'origin/master' by 7 commits.

(use "git push" to publish your local commits)

**4）在master分支上创建临时分支修复bug：**

$ git checkout -b issue-101

Switched to a new branch 'issue-101'

**5）修复完bug，添加、提交：**

Administrator@Helay-computer MINGW64 ~/Desktop/gitskills (issue-101)

$ git add test.txt

Administrator@Helay-computer MINGW64 ~/Desktop/gitskills (issue-101)

$ git commit -m "fix bug 101"

[issue-101 158237c] fix bug 101

1 file changed, 1 insertion(+)

**6）切换到master分支上：**

$ git checkout master

Switched to branch 'master'

Your branch is ahead of 'origin/master' by 7 commits.

(use "git push" to publish your local commits)

**7）合并分支：**

$ git merge issue-101

Updating d9b26da..158237c

Fast-forward

test.txt | 1 +

1 file changed, 1 insertion(+)

**8）删除bug分支：**

$ git branch -d issue-101

Deleted branch issue-101 (was 158237c).

**9）回到原来的dev分支上，做之前没有做完的工作**

$ git checkout dev

Switched to branch 'dev'

**10）查看状态：**

$ git status

On branch dev

nothing to commit, working tree clean

**工作区干净，原来的没有做完的工作被隐藏了**

**11）使用git stash list查看隐藏的内容：**

$ git stash list

stash@{0}: WIP on dev: c439e63 add aaaa

stash@{1}: WIP on dev: c439e63 add aaaa

**12）删除隐藏内容同时恢复隐藏内容为可见状态：**

$ git stash pop

On branch dev

Changes not staged for commit:

(use "git add <file>..." to update what will be committed)

(use "git checkout -- <file>..." to discard changes in working directory)

modified: test.txt

no changes added to commit (use "git add" and/or "git commit -a")

Dropped refs/stash@{0} (bc706ad5aa61c27839b2e01e4caf08288a6d6048)

（其中git stash pop相当于git stash apply和git stash drop两个命令的功能，即删除同时恢复）

**13）继续原来的工作**