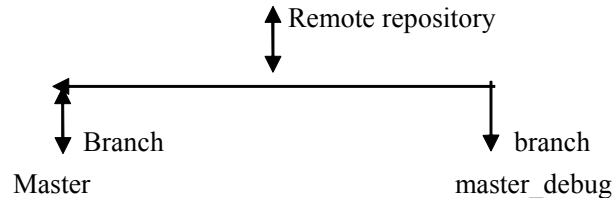


KTC git 工作处理方法

一、工作流程

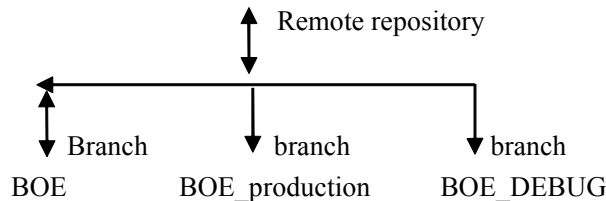
1、不论是改 KTC 公版 BUG，还是客户专用软件 BUG 都开分支来修改，修改完后对比两个分支，确认是否有误添加的地方。KTC 公版 BUG 从，KTC 公版开分支，客户专用软件 BUG 从客户专用分支开分支；



Master 可做 PUSH, PULL, MERGE

Master_debug, 只修改测试，完后删除；

2、生产软件制作每天从远程仓储 fetch, merge 数据到本地，再配置软件。只做生产的分支不能做 PUSH 命令；



BOE 可做 PUSH, PULL, MERGE

BOE_production 只做 PULL 或 fetch, MERGE

BOE_DEBUG 只修改测试，完后删除

二、常用命令

1、从远程拷贝代码到本地

```
git clone git@192.168.8.72:6a608-0312/ics.git
```

2、切换到工作分支

```
git checkout -b temp //新建并切换到工作分支，如改某个 BUG
```

```
git checkout KTC //切换到客户专用分支工作，如果是改某个客户的 BUG，则从客户分支上再开分支，不在已量产的分支代码上直接修改
```

3、删除工作分支

```
git branch -d branch1 //删除 branch1 分支
```

4、将修改的问题做暂存

```
git add -i //交互式，按提示命令选择相关操作
```

```
git add . //暂存所有修改文件
```

5、撤销某个暂存文件

```
git reset files
```

6、将暂存文件提交

```
git commit -m " " //将文件提交到本地保存
```

7、撤销某个提交

```
git reset --soft/mixed/hard commit_id //soft,mixed,hard 差异百度一下你就知道了
```

8、将处理问题对策合并

- A.系统 BUG: `git merge temp` //将当前工作分支切换到 `master`, 再 `merge` 修改 BUG 的分支
 - B.客户专用软件 BUG:`git merge temp` //将当前分支切换到客户专用分支, 再 `merge` 修改 BUG 的分支
- 9、将合并后的代码提交
- A.提交系统 BUG: `git push origin master` //首先将工作分支切换到 `master` 分支, 再做 `PUSH`;
 - B.提交客户 BUG: `git push origin KTC` //首先将工作分支切换到 `KTC` 分支, 在做 `PUSH`;
- 10、从远程获取数据到本地
- A. `git fetch`
 - B. `git fetch origin branch1`
- 11、将从远程获取到的数据与本地做合并
- A.`git merge origin/master` //将从远程 `master` 获取到本地的数据合并到当前分支
 - B.`git merge origin/branch1` //将从远程 `branch1` 获取到本地的数据合并到当前分支
- 11、查看 2 个版本的差异
- A.`git diff commit_id1 commit_id2` //比较 `commit_id1` `commit_id2` 的差异
 - B.`git diff --staged` //查看 `stage` 内容的差异
 - C.`git diff` //查看最新修改还没放到 `stage` 中的差异
 - D.`git diff filename` //查看指定文件的修改
 - E.`git diff dirname/` //查看指定目录的修改
 - F.`git diff branch1 branch2` //查看 2 个分支的差异