

深圳韵达速递有限公司

# 协作开发规范说明

---

## 第二版

信息与流程管理部  
开发组

本文主要针对多人开发进行规范，从代码规范、命名规范、版本控制规范、项目整体流程规范四个方面进行说明，并对其中的过程进行详细的解释。

张顺 2016 年 9 月 6 日 11:20:19

# 协作开发规范说明

**摘要：**本文主要针对多人开发进行规范，从代码规范、命名规范、版本控制规范、项目整体流程规范四个方面进行说明，并对其中的过程进行详细的解释。

**关键字：**协作；开发；规范

# 目 录

第一章 代码规范 .....	4
1.1 action 部分 .....	4
1.2 service 部分 .....	4
1.3 dao 部分 .....	4
1.4 实体类 .....	4
1.5 jsp 等前端部分 .....	4
第2章 命名规范 .....	5
2.1 数据库、表命名 .....	5
2.2 类命名 .....	5
2.3 Jsp、html、js、css 命名 .....	5
2.4 文件夹命名 .....	5
2.5 包命名 .....	5
2.6 方法命名 .....	5
2.7 变量命名 .....	5
2.8 常量命名 .....	5
第3章 版本控制规范 .....	6
3.1 获取原作者版本库分支 .....	6
3.2 克隆远端版本库到本地 .....	7
3.3 从远端获取最新的版本库 .....	8
3.4 保存本地改动与上传改动到远端版本库 .....	9
3.5 本地回滚与远端回滚 .....	12
3.5.1 本地回滚 .....	12
3.5.2 远端回滚 .....	14
3.6 远端版本库与原作者同步 .....	15
3.7 发送改动给原作者 .....	18

3.8 版本控制日常操作流程.....20

第4章 项目整体流程规范.....22

4.1 项目整体流程.....22

4.2 需求分析.....23

4.3 系统设计.....24

4.4 开发排期.....24

附 录.....25

术语定义.....25

## 第一章 代码规范

### 1.1 action 部分

action 须继承 **MyBaseAction**，实现 **IMyBaseAction**。一般情况在里面的方法里编写代码（每个方法都有注释说明），尽量不要使用自定义方法。

### 1.2 service 部分

当 action 中有重复代码，并且可以提取出来的情况下，在 service 中编写方法，并在 service 的接口中添加改方法，并做好注释。

### 1.3 dao 部分

Dao 已经完善，基本上不会有改变。

### 1.4 实体类

采用 MyEclipse 的生成功能，当情况需要可以对实体类进行更改，但是更改的原则是：只能新增，不能对原有内容进行修改或删除。

### 1.5 jsp 等前端部分

Jsp 采用之前编写的模板进行编写，采用 EL 表达式操作数据，如果有重复的 js 或 css 代码可以提取出，写在 `FRAMEWORK/js/js.js` 或 `FRAMEWORK/css/css.css` 文件中，采用类选择器或 ID 选择器，不要使用类选择器（除非特殊情况），原则是不要对其他页面产生影响。

## 第 2 章 命名规范

### 2.1 数据库、表命名

首字母小写，多个单词采用下划线命名法<sup>[2]</sup>。

### 2.2 类命名

首字母大写，多个单词用采用驼峰法命名<sup>[1]</sup>规则。

### 2.3 Jsp、html、js、css 命名

首字母小写，多个单词采用下划线命名法<sup>[2]</sup>。

### 2.4 文件夹命名

首字母小写，多个单词采用下划线命名法<sup>[2]</sup>。特殊文件夹全部大写。

### 2.5 包命名

尽量用单个单词，实在为多个单词可以用简称或者驼峰法命名<sup>[1]</sup>，首字母小写

### 2.6 方法命名

首字母小写，驼峰法<sup>[1]</sup>。

### 2.7 变量命名

首字母小写，一般情况使用驼峰法<sup>[1]</sup>，特殊情况可以采用下划线命名法<sup>[2]</sup>。

### 2.8 常量命名

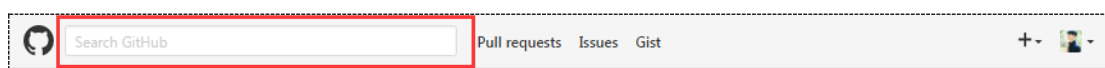
全部大写，多个单词采用“\_”连接。

## 第 3 章 版本控制规范

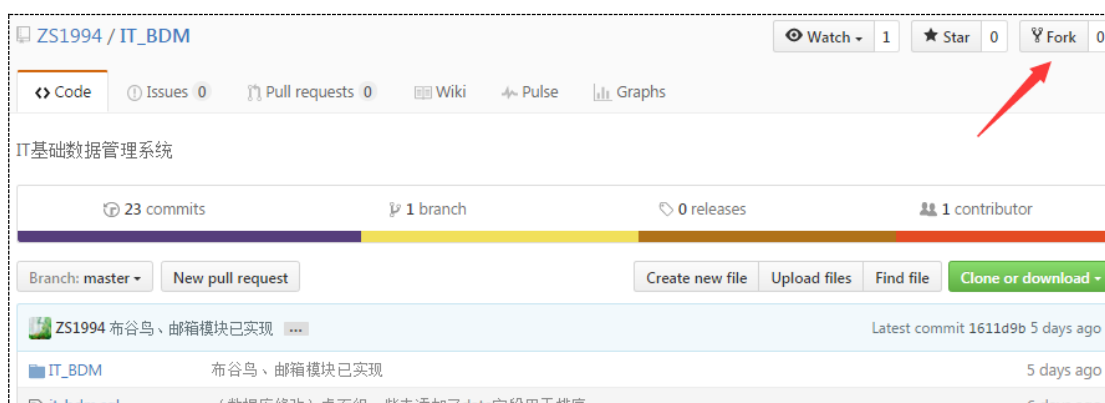
使用 Git 进行版本管理、文件交换，版本合并等等操作。

### 3.1 获取原作者版本库分支

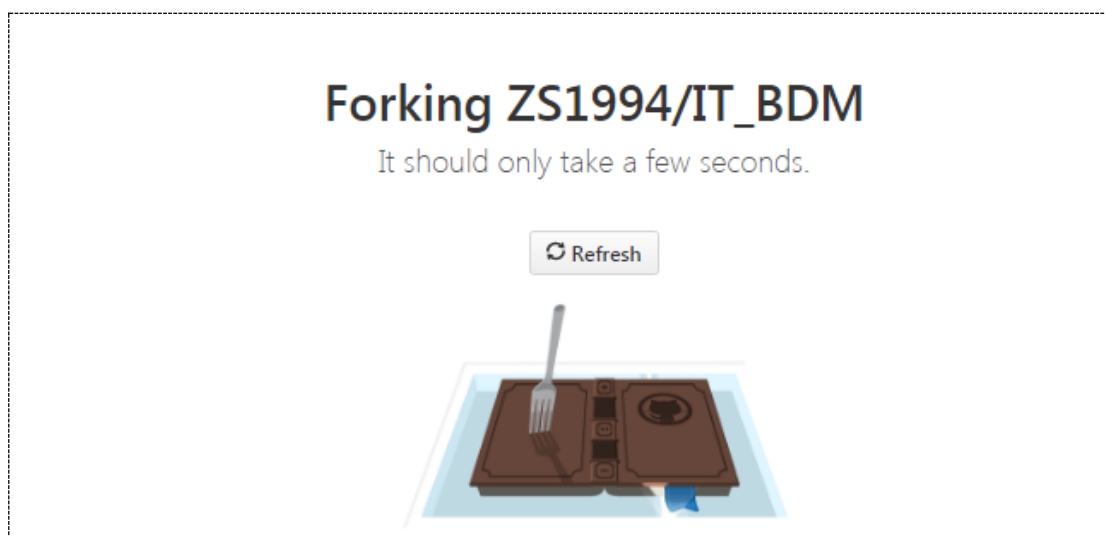
输入项目名称，搜索项目。



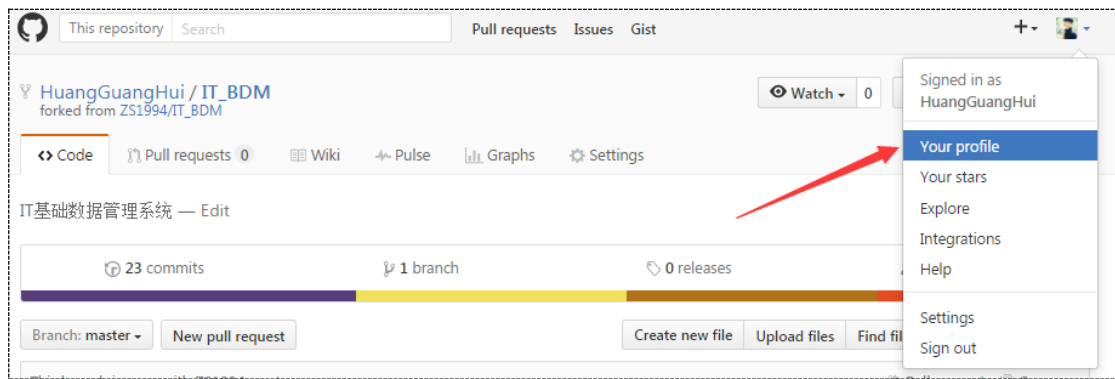
点击 Fork 按钮。



等待 fork 成功。



成功之后，点击进入自己的资料。



可以看到刚刚 fork 的库。

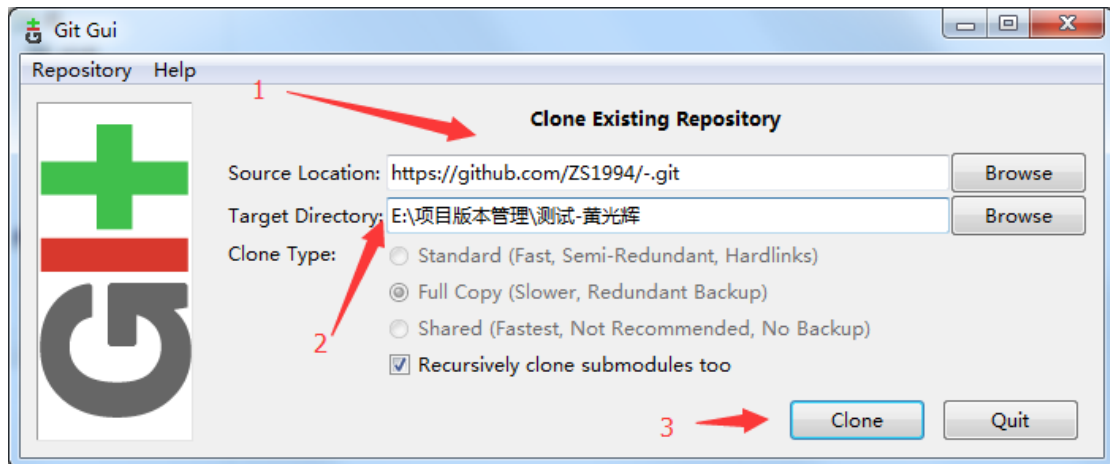


### 3.2 克隆远端版本库到本地

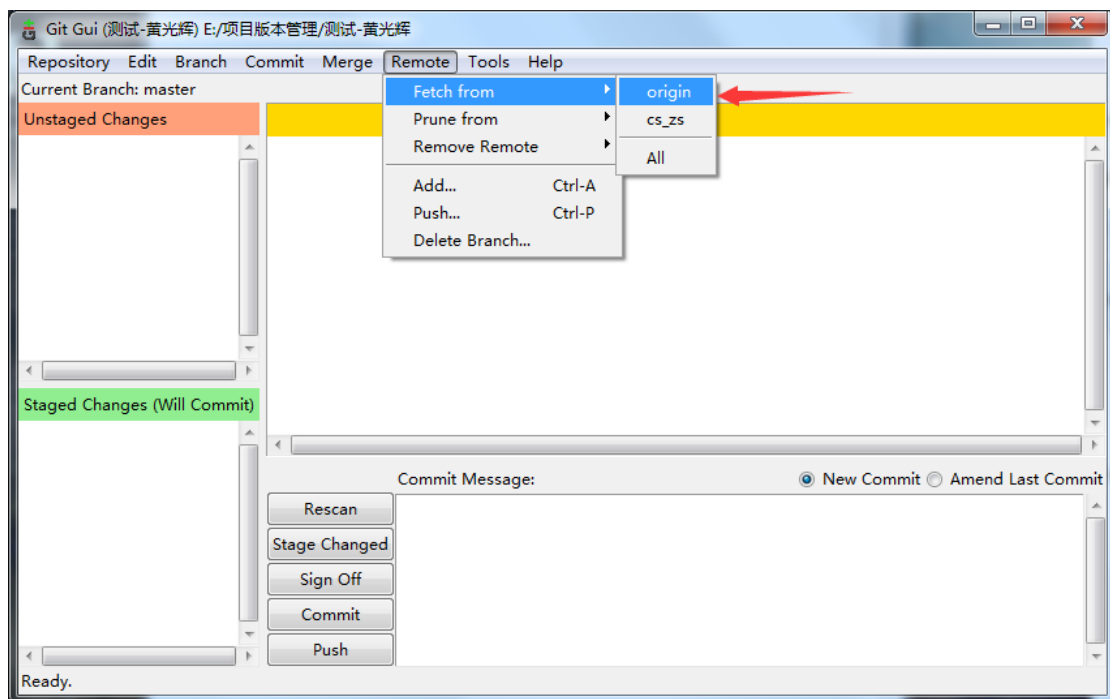


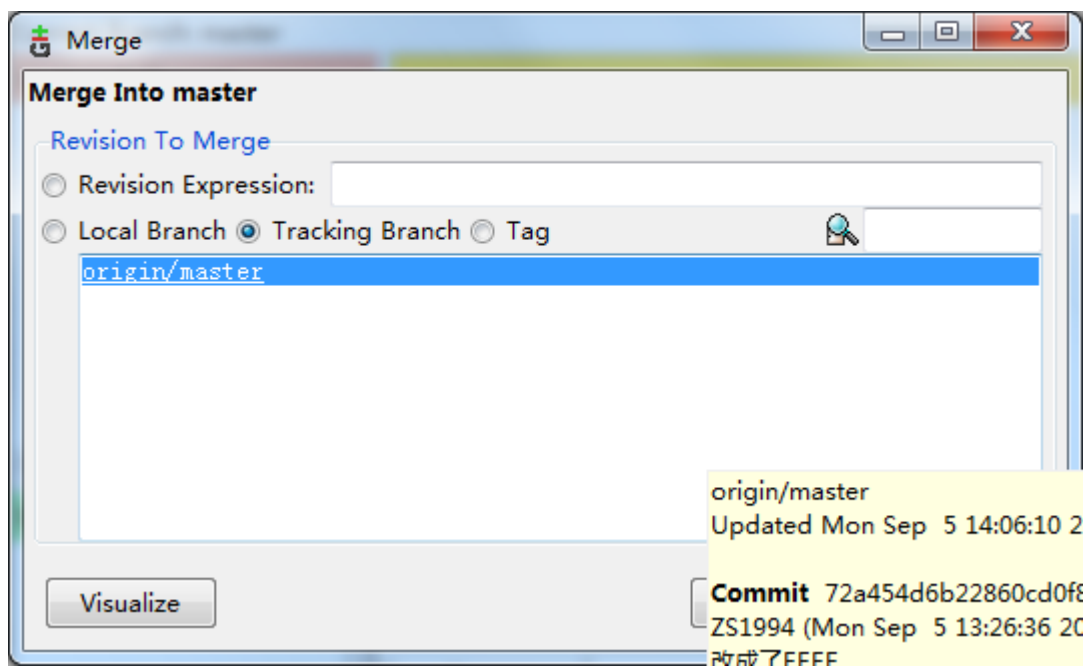
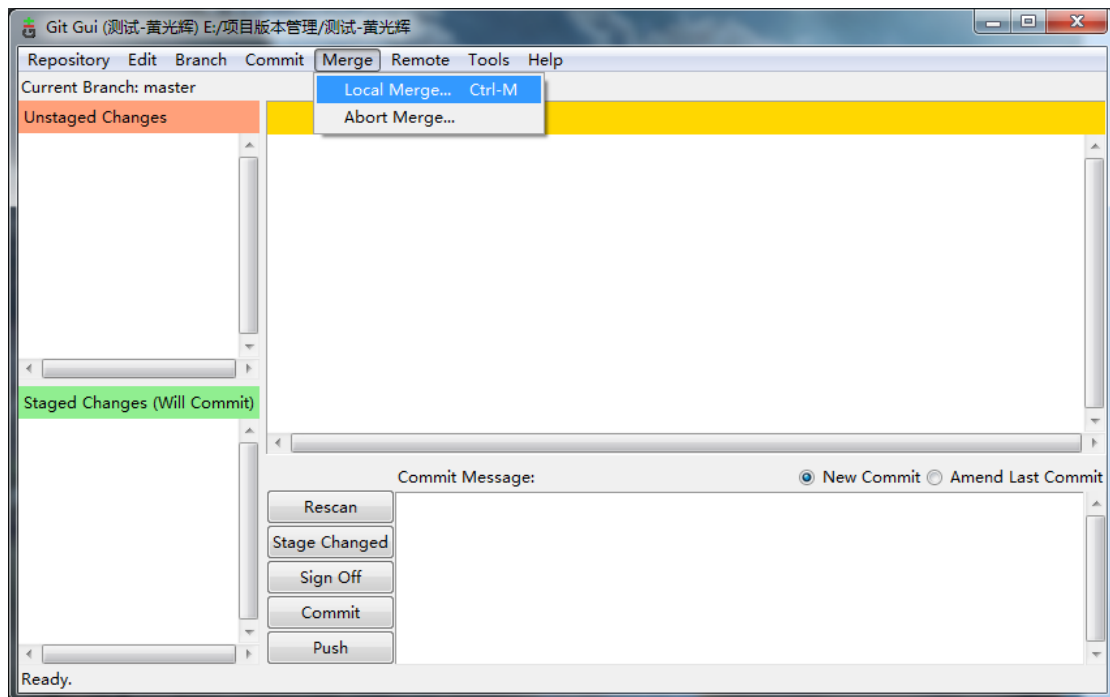


- 1 填写 github 的地址(比如我的那个 `https://github.com/ZS1994/-.git` ),
- 2 填写将要创建的本地库路径, 其他选项默认, 3 点击 clone。



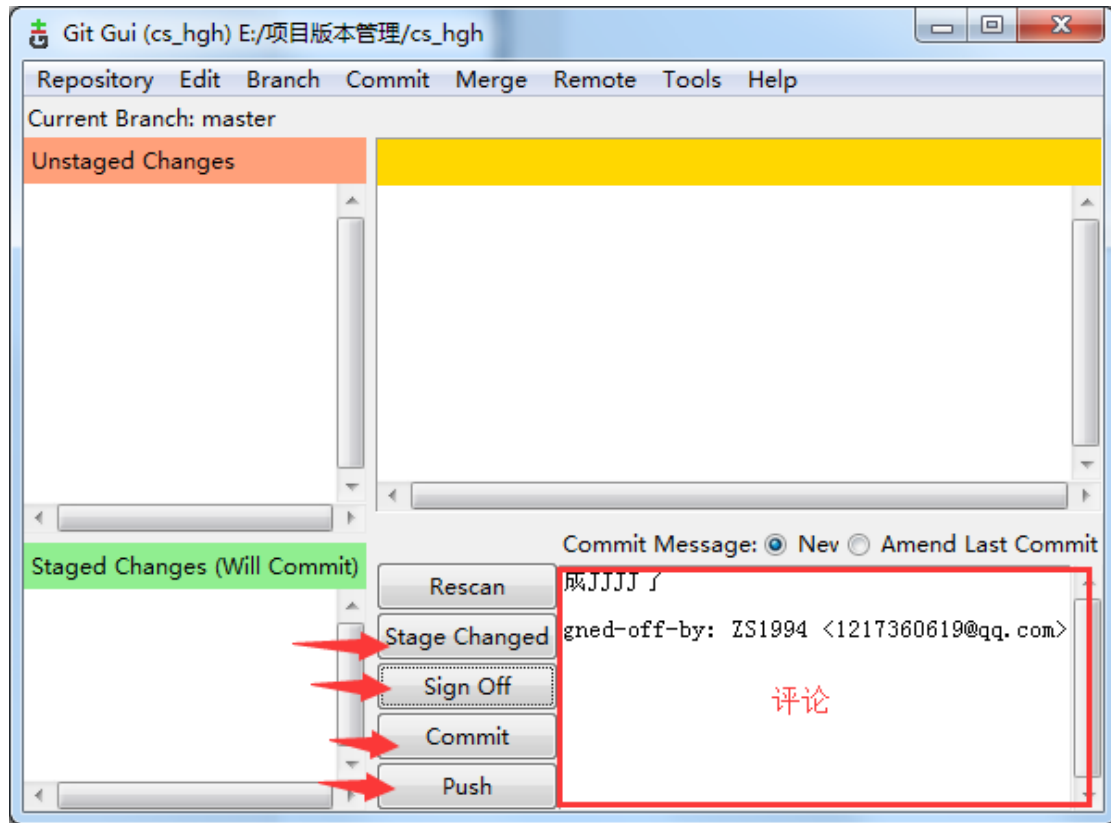
### 3.3 从远端获取最新的版本库



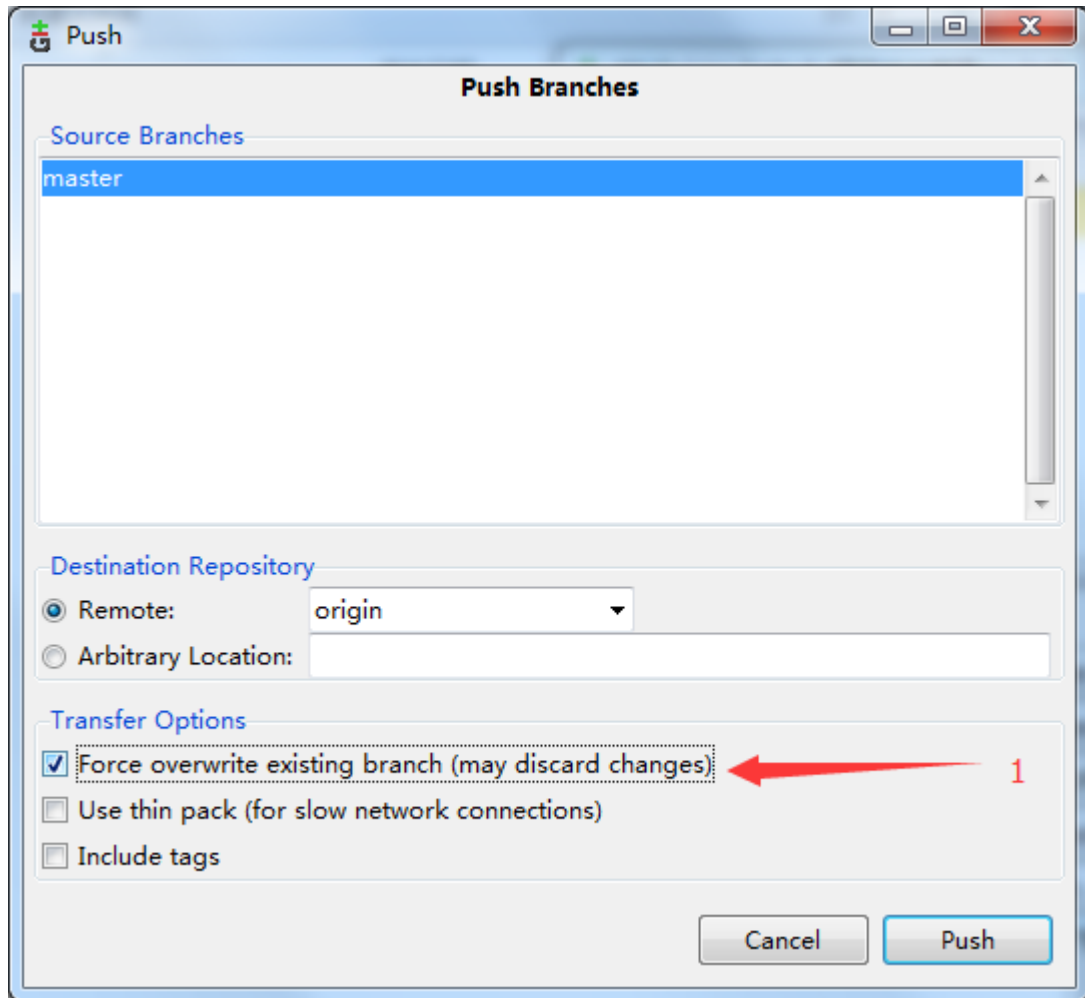


### 3.4 保存本地改动与上传改动到远端版本库

第一步写评论，第二点击 stage Changed 缓存改动，第三步点击 sing off 签名，第四步点击 commit 提交改动，第五步点击 push 上传改动。

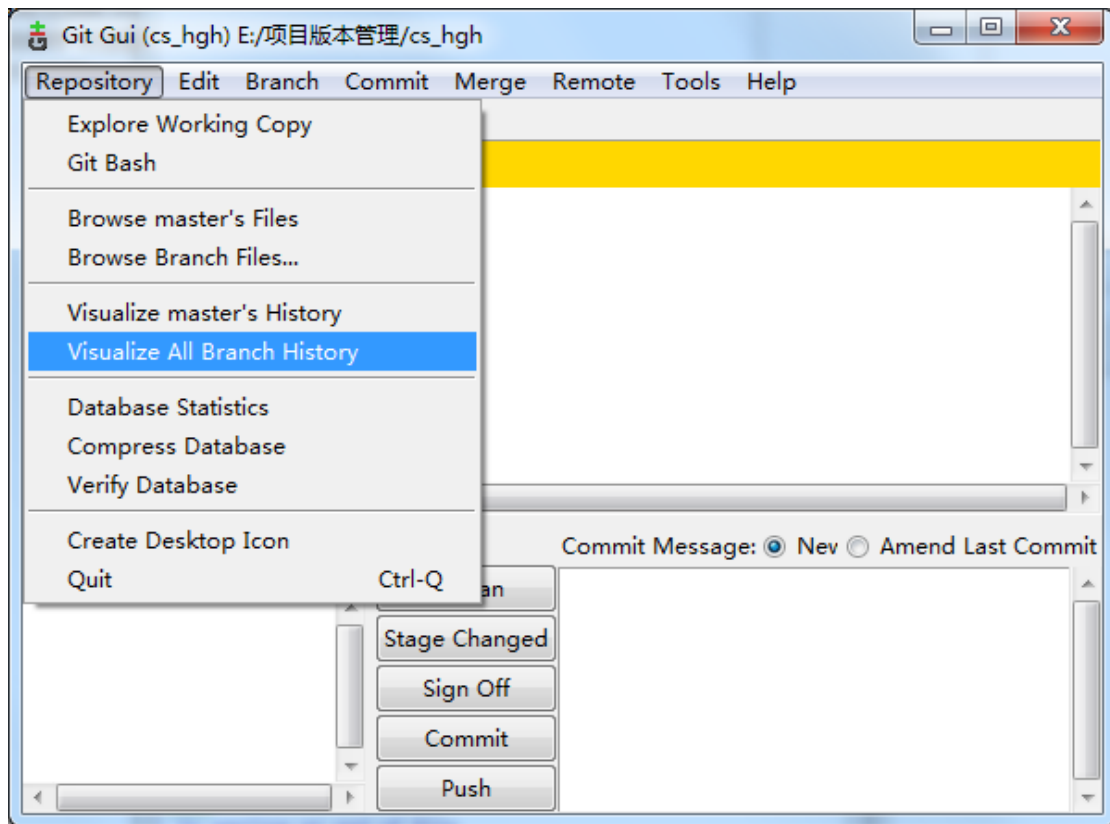


1 注意这里一定要打勾，点击 push，输入账号密码完成上传。

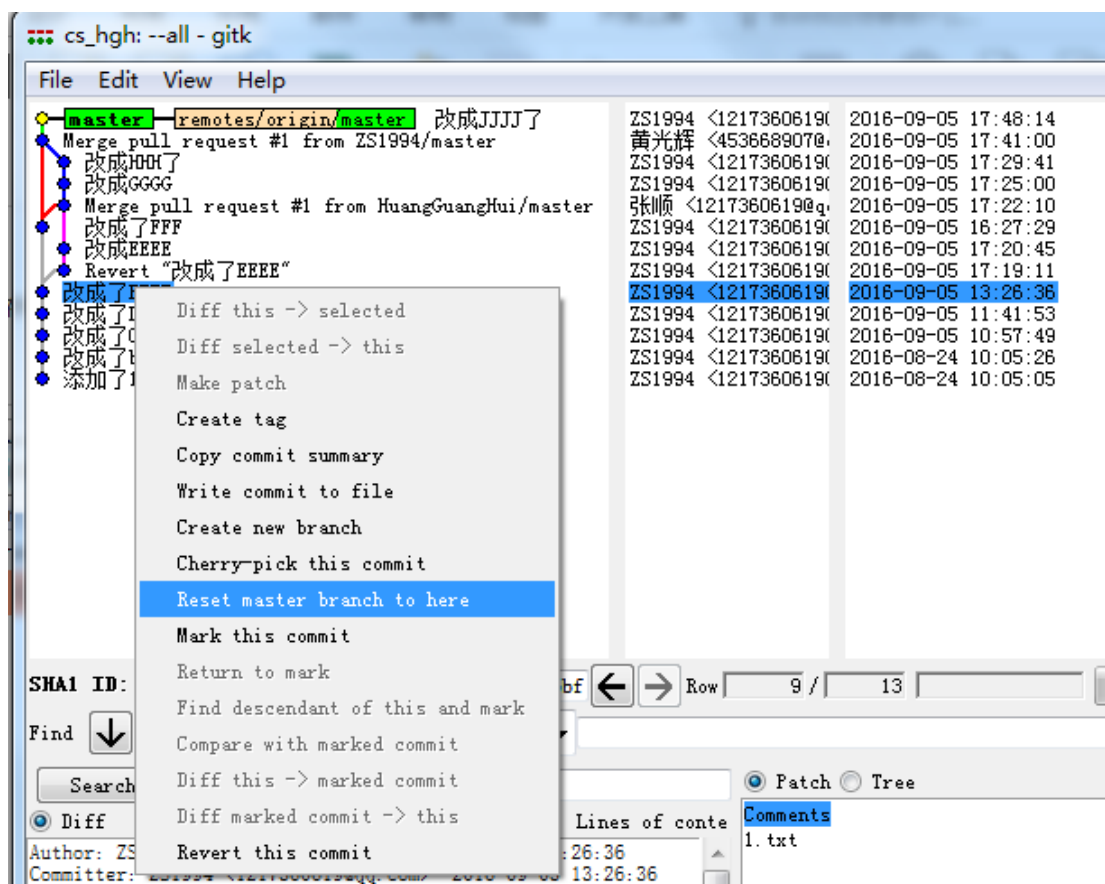


### 3.5 本地回滚与远端回滚

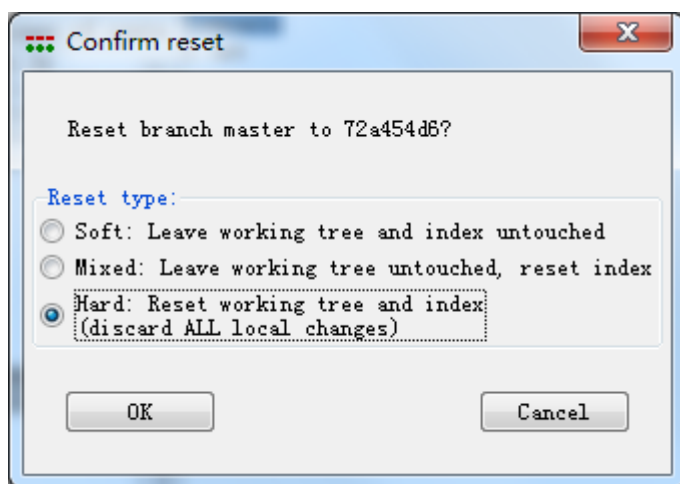
#### 3.5.1 本地回滚



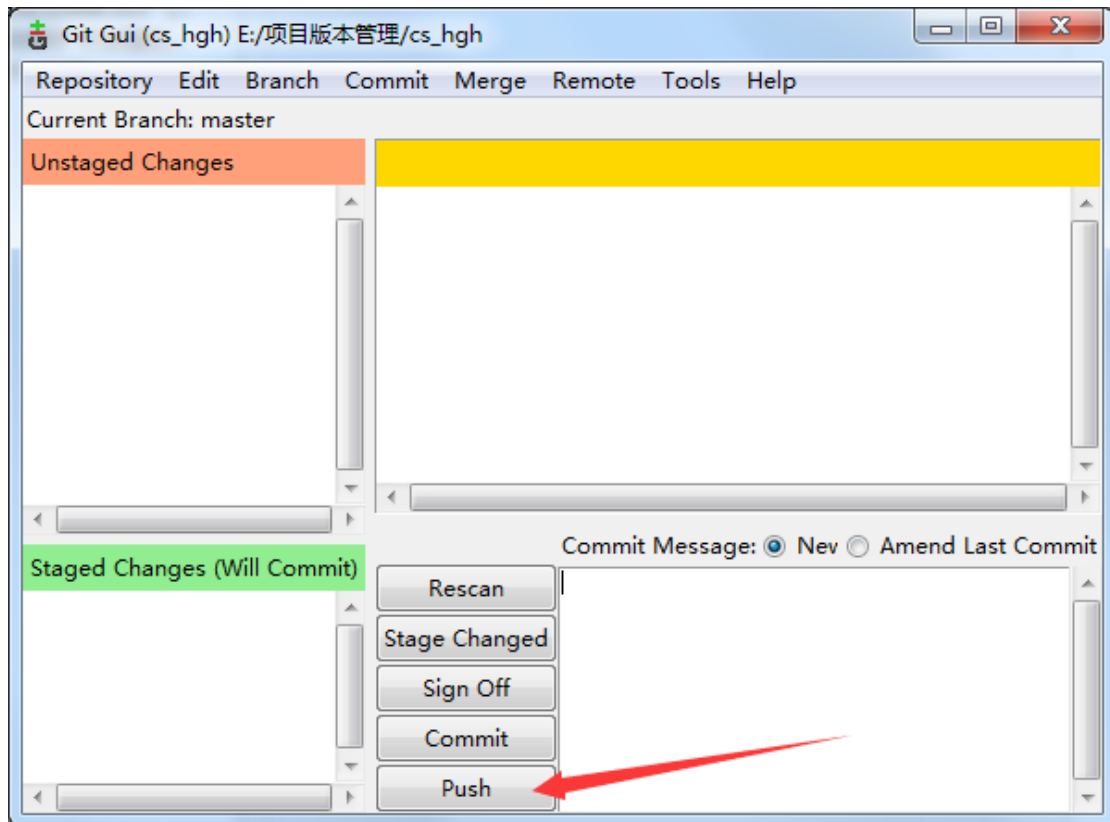
右键要回滚的版本，选择该选项。



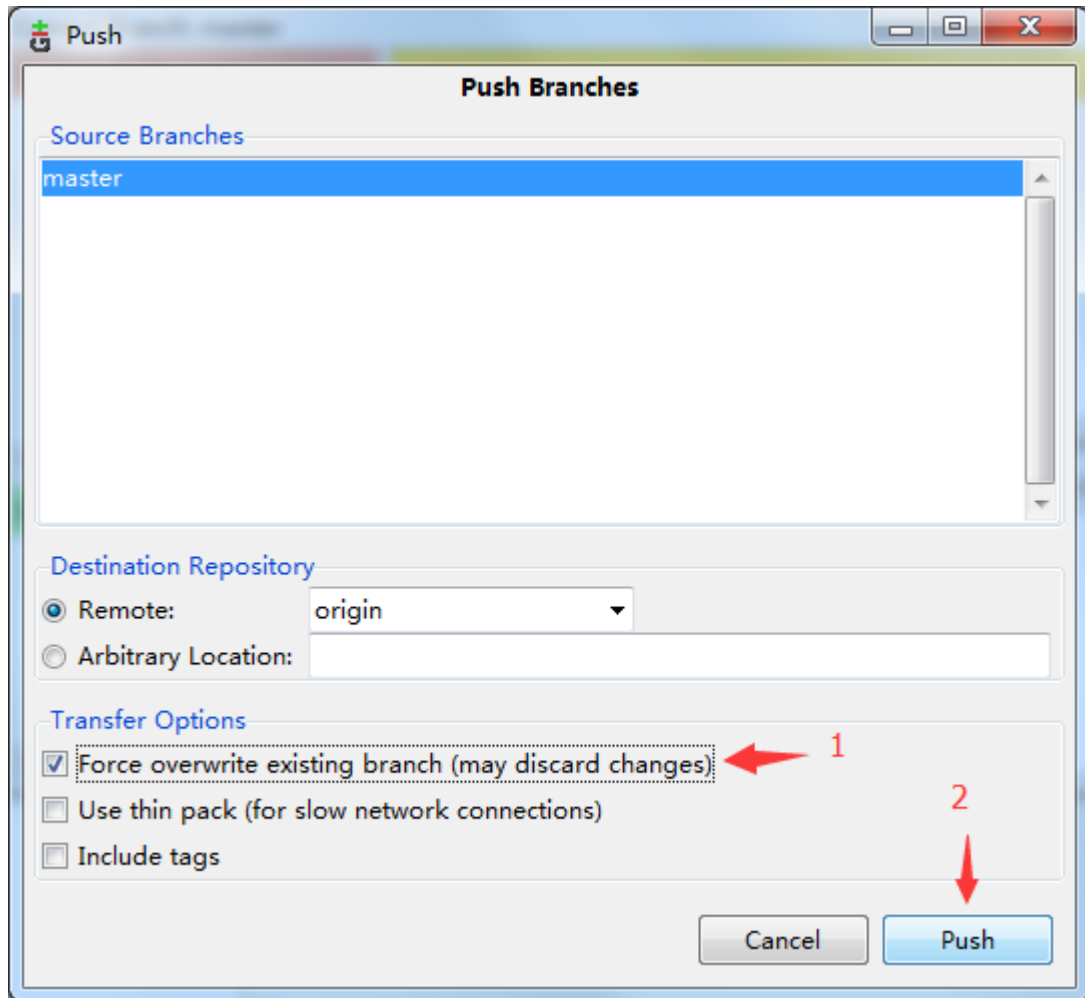
选择 hard 模式回滚，本地回滚就成功了，但是远端的版本依然没变。



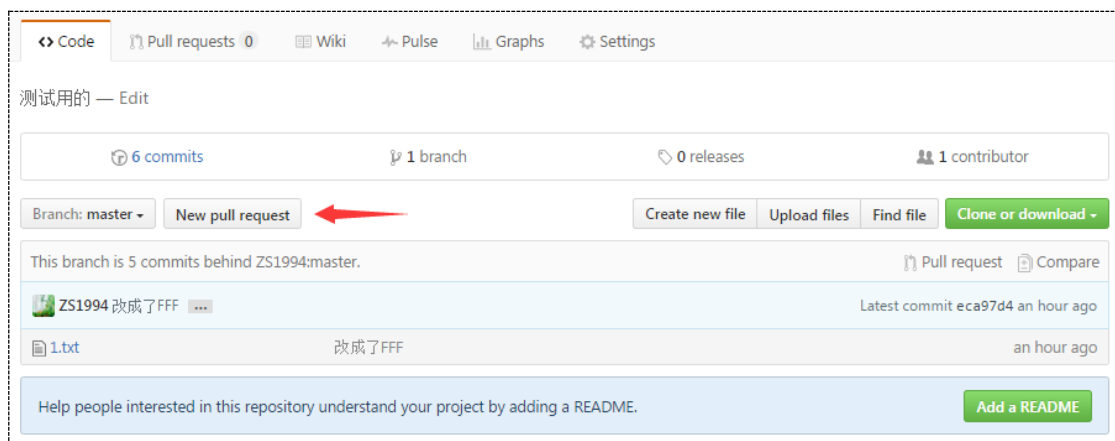
### 3.5.2 远端回滚



1 注意打勾, 2 点击 push 提交, 输入账号密码就能将远端的版本库也回滚了。



### 3.6 远端版本库与原作者同步





## Comparing changes

Choose two branches to see what's changed or to start a new pull request. If you need to, you can also [compare across forks](#).

base fork: HuangGuangHui/cs\_zs ▾ base: master ▾ ... head fork: ZS1994/cs\_zs ▾ compare: master ▾

✓ Able to merge. These branches can be automatically merged.

[Create pull request](#) Discuss and review the changes in this comparison with others. ⓘ

↔ 5 commits    1 file changed    0 commit comments    2 contributors

Commits on Sep 05, 2016

↗	ZS1994	<a href="#">Revert "改成了EEEE" ...</a>	31372f
↗	ZS1994	<a href="#">改成EEEE ...</a>	1f8e35
↗	ZS1994	<a href="#">Merge pull request #1 from HuangGuangHui/master ...</a>	790084
↗	ZS1994	<a href="#">改成GGGG ...</a>	8c0873

## Comparing changes

Choose two branches to see what's changed or to start a new pull request. If you need to, you can also [compare across forks](#).

base fork: ZS1994/cs\_zs ▾ base: master ▾ ... head fork: HuangGuangHui/cs\_zs ▾ compare: master ▾

**There isn't anything to compare**

ZS1994:master is up to date with all commits from HuangGuangHui:master. Try [switching the base](#) for your comparison.


1 输入标题，2 输入评论，3 点击按钮。

## Open a pull request

Create a new pull request by comparing changes across two branches. If you need to, you can also [compare across forks](#).

base fork: **HuangGuangHui/cs\_zs** base: **master** ... head fork: **ZS1994/cs\_zs** compare: **master**

✓ **Able to merge.** These branches can be automatically merged.

 Title 1

Write Preview AA B i “ < > ↻ ⋮ ⋮ ⋮ ↶ @

Leave a comment 2

Attach files by dragging & dropping, [selecting them](#), or pasting from the clipboard.

Styling with Markdown is supported


3 [Create pull request](#)

[Code](#) **[Pull requests](#) 1** [Wiki](#) [Pulse](#) [Graphs](#) [Settings](#)


### 同步测试 #1











[Open](#) **HuangGuangHui** wants to merge 5 commits into **HuangGuangHui:master** from **ZS1994:master**


[Conversation](#) 0 [Commits](#) 5 [Files changed](#) 1

 **HuangGuangHui** commented just now Owner + 😊 ✎

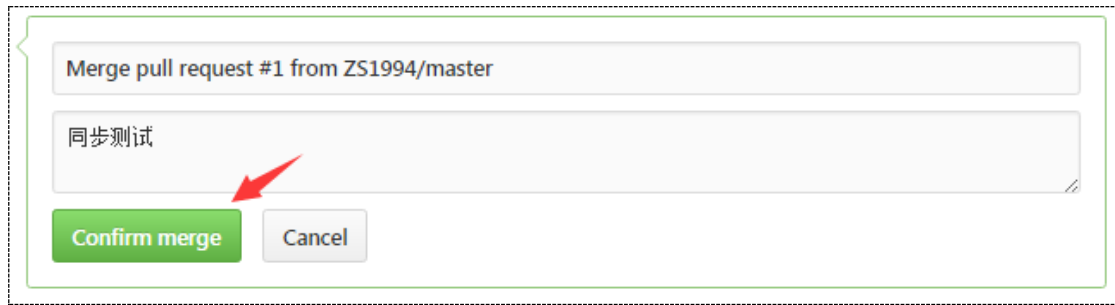
这是同步测试的测试

 **ZS1994** added some commits 21 minutes ago

-   Revert "改成了EEEE" ... 31372ff
-   改成EEEE ... 1f8e35a
-   Merge pull request #1 from HuangGuangHui/master ... 7900843
-   改成GGGG ... 8c08734
-   改成HHH了 607f006

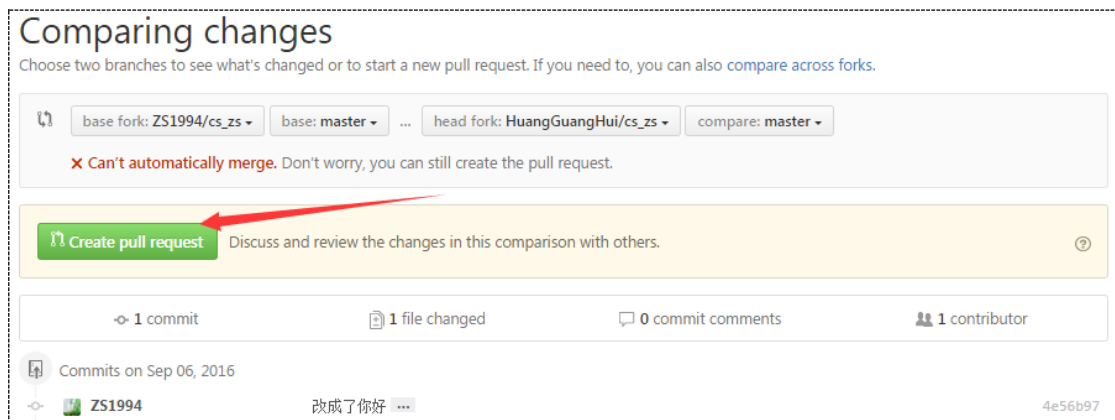
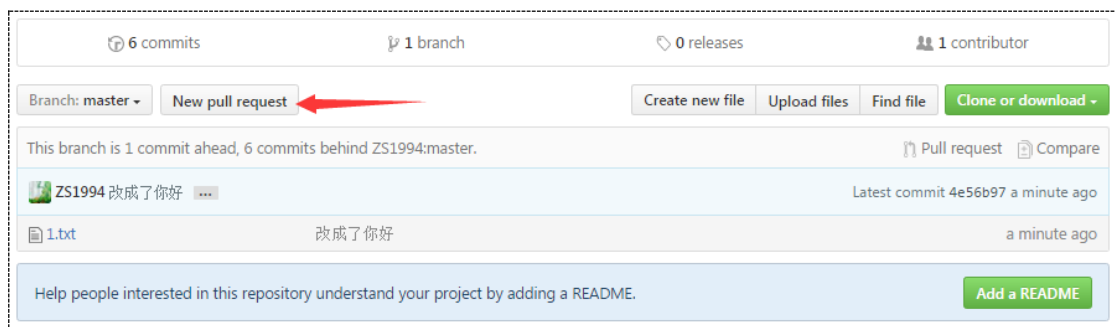
 ✓ **This branch has no conflicts with the base branch**  
Merging can be performed automatically.

[Merge pull request](#) ↗ You can also open this in [GitHub Desktop](#) or view [command line instructions](#).



### 3.7 发送改动给原作者

点击 New pull request 按钮。



## Open a pull request

Create a new pull request by comparing changes across two branches. If you need to, you can also [compare across forks](#).

base fork: ZS1994/cs\_zs ▾ base: master ▾ ... head fork: HuangGuangHui/cs\_zs ▾ compare: master ▾

✖ Can't automatically merge. Don't worry, you can still create the pull request.



改成了你好

Write

Preview

AA ▾ B i

“ ” < > ↻

☰ ☰ ☰

↶ @ 🚩

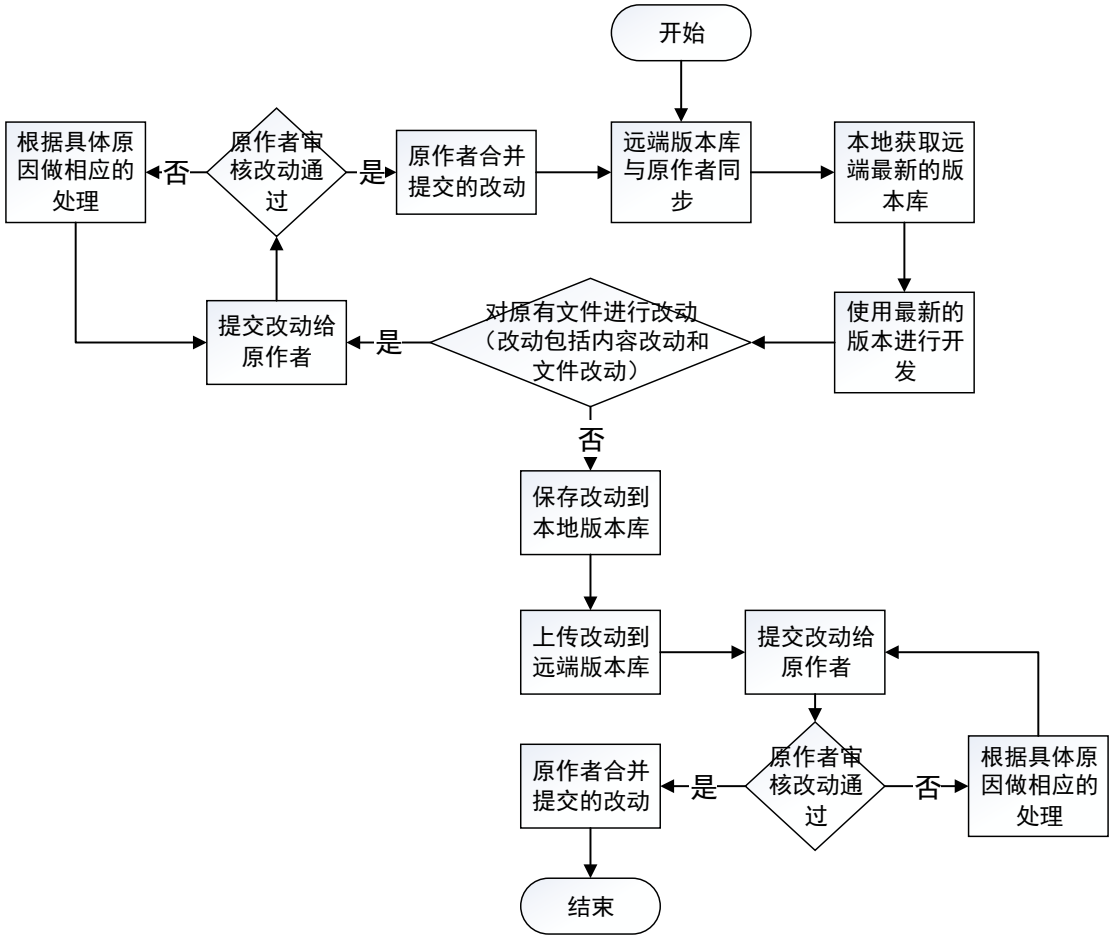
Signed-off-by: ZS1994 <1217360619@qq.com>

Attach files by dragging & dropping, [selecting them](#), or pasting from the clipboard.

📖 Styling with Markdown is supported

Create pull request

3.8 版本控制日常操作流程



其中“对原有文件进行改动”这一步为何要做这个判断，因为根据测试结果如果对原有文件进行改动，那么就无法将两人的版本进行合并，以下是测试表。

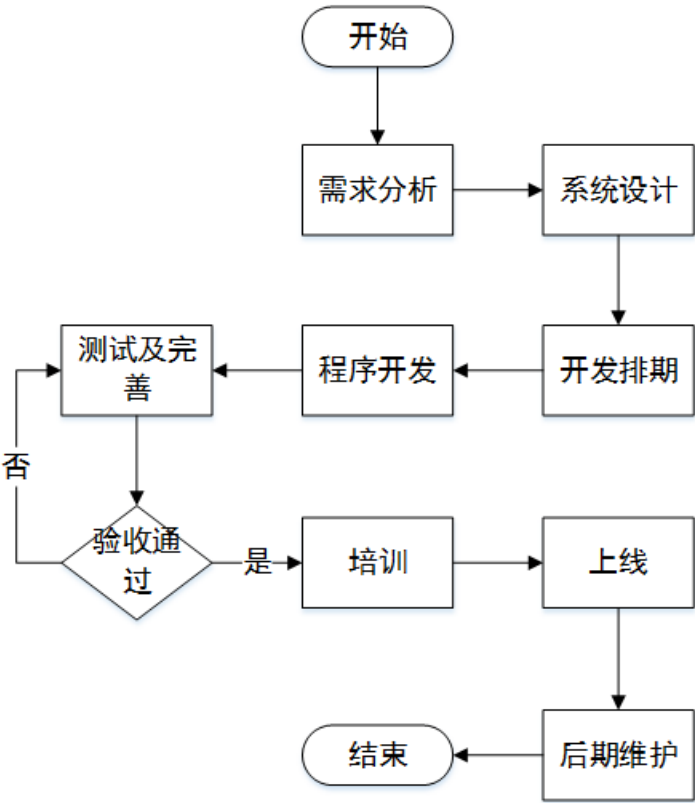
版本控制测试表

场景	结果
双方的原有文件不改动，各添加不同的文件。	成功合并
双方的部分原有文件不变，部分原有文件内容各自进行不同的改动。	无法合并
双方部分原有文件各自新增不同的内容，双方都没有新增文件。	无法合并
一方原有文件不变，另一方新增文件，并对原有文件进行改动。	成功合并

根据测试结果可得出，双方协作开发时必须保证：1，双方版本一致；2，原有内容不可更改，一旦有更改，双方一定要先做同步工作，然后再继续进行开发。

## 第 4 章 项目整体流程规范

### 4.1 项目整体流程



## 4.2 需求分析

系统组与需求方和用户讨论并最终确定系统需求分析说明书，文档需包含以下内容：

- 概述
  - 目的
  - 范围
  - 用户群体
  - 术语定义
- 系统说明
  - 总体功能分析
    - ◆ 功能概括图
    - ◆ 功能用例图
  - 系统中的用户与角色
- 功能性需求
  - 系统模块
    - ◆ 业务描述
    - ◆ 使用者
    - ◆ 输入要素
    - ◆ 处理流程
    - ◆ 输出要素
- 非功能性需求
  - 技术需求
  - 开发时间需求
  - 文档需求
- 验收标准
- 需求确认



### 4.3 系统设计

开发组内部讨论并制定系统设计方案，系统设计方案包含以下内容：

- 技术选型
  - 框架选择
  - 数据库选择
- 架构设计
  - 系统层级模型
  - 前端布局方案
  - 底层接口设计

### 4.4 开发排期

开发组内部制定开发排期表，需包含以下内容：

- 项目通讯录
- 里程碑
- 时间表
- 各方签核

## 附 录

### 术语定义

[1]驼峰命名法：当变量名或函数名是由一个或多个单词连结在一起，而构成的唯一识别字时，第一个单词以小写字母开始；第二个单词的首字母大写。

[2]下划线命名法：采用下划线分割小写字母的方式命名。命名应当准确，不引起歧义，且长度适中。