## Git客户端安装及使用说明

1. 安装git

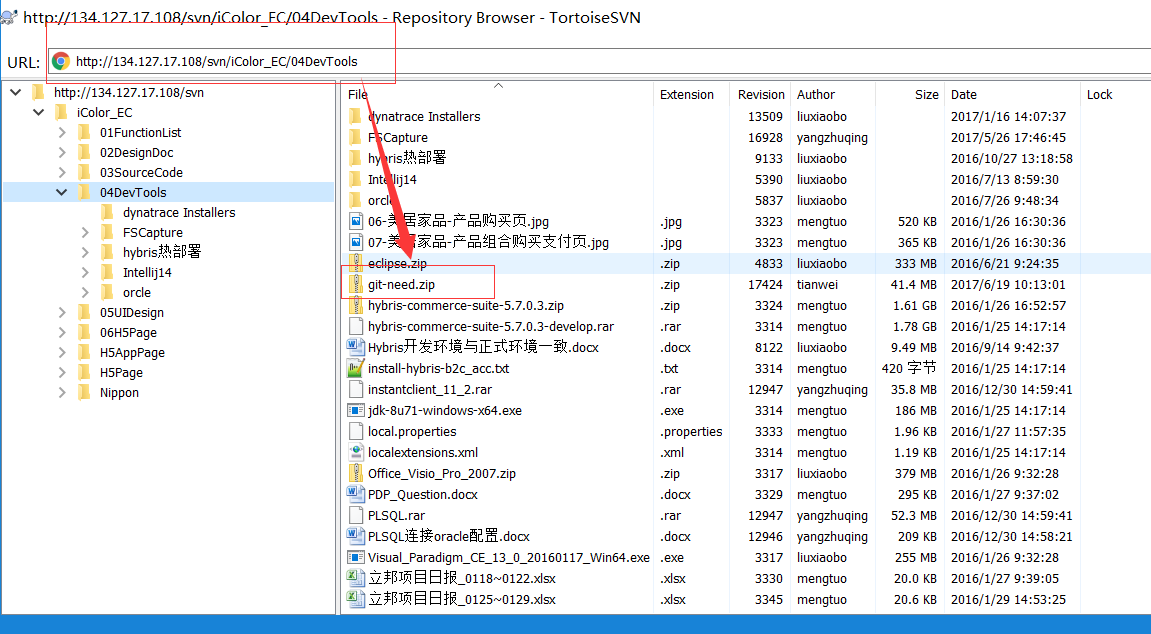
官网下载安装git，下载地址：<https://git-scm.com/>，或者从svn服务器上下载已经存在的客户端。

1. 安装git图形化操作工具sourceTree

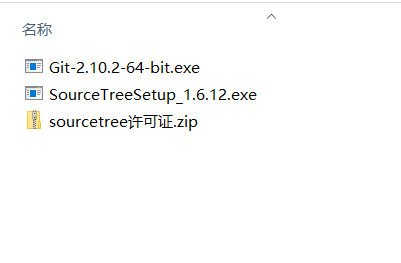
Git默认的操作方式是通过命令行的方式进行操作，为方便开发我们使用图形化操作工具。可自行下载，或者使用svn服务器上下载的版本。

服务器下载git所需要工具地址：

http://134.127.17.108/svn/iColor\_EC/04DevTools/git-need.zip



下载减压之后，有3个文件，一个是git安装包，先使用这个进行git安装。安装过程与其它windows程序一致，直接下一步即可，无任何配置需要更改。



安装git完成之后，在任意文件夹点击右键，可以出现以下结果：



继续安装git图形化工具sourceTree。该软件使用进行注册，注册过程中，需要连接到国外网站获取验证码。大家可以自行去注册，或者使用已经准备好的license进行注册。

我们使用**SSH**进行访问远程服务器，所以需要在本地生成密钥，然后将公钥上传到服务器。

操作如下：

2.1客户端生成RSA密钥。

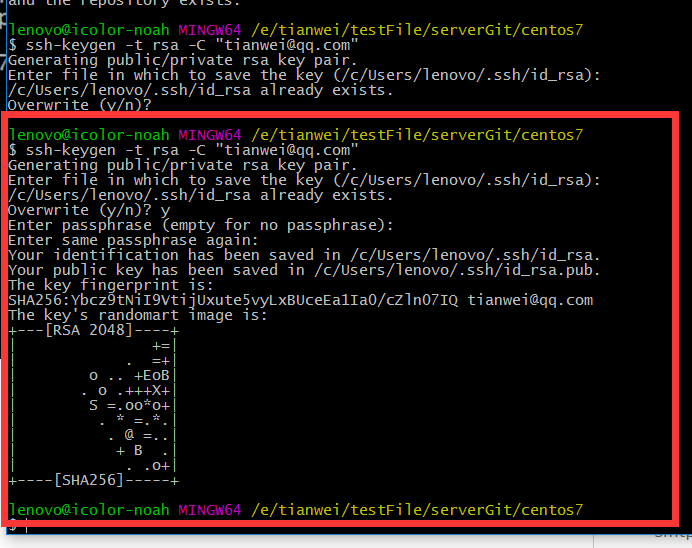
在任何一个文件夹中，右键打开git Bash here。输入以下代码：

ssh-keygen -t rsa –C “自己的公司邮箱”

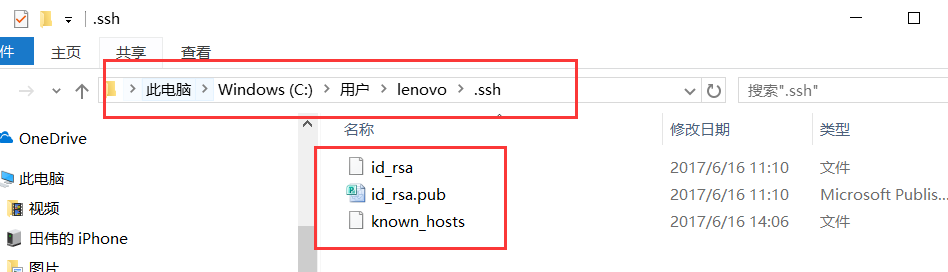
例：

|  |
| --- |
| ssh-keygen -t rsa -C "TianWeiSH@nipponpaint.com.cn" |

然后一直回车



在相应位置找到刚刚生成的密钥文件：



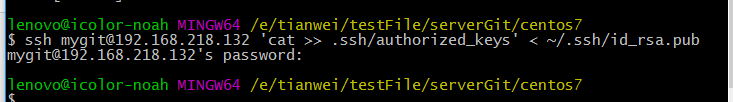
上传公钥到服务器：

执行代码：

|  |
| --- |
| ssh icolorgit@192.168.13.50 'cat >> .ssh/authorized\_keys' < ~/.ssh/id\_rsa.pub |

回车，需要输入icolorgit用户的密码： a.o7a@sh

操作示例如下：



1. sourceTree使用说明

安装注册好的sourceTree应当出现以下界面：



* 1. 克隆远程git仓库

git远程仓库地址格式为 ：<https://github.com/xxxx/helloworld.git>，地址结尾是.git。

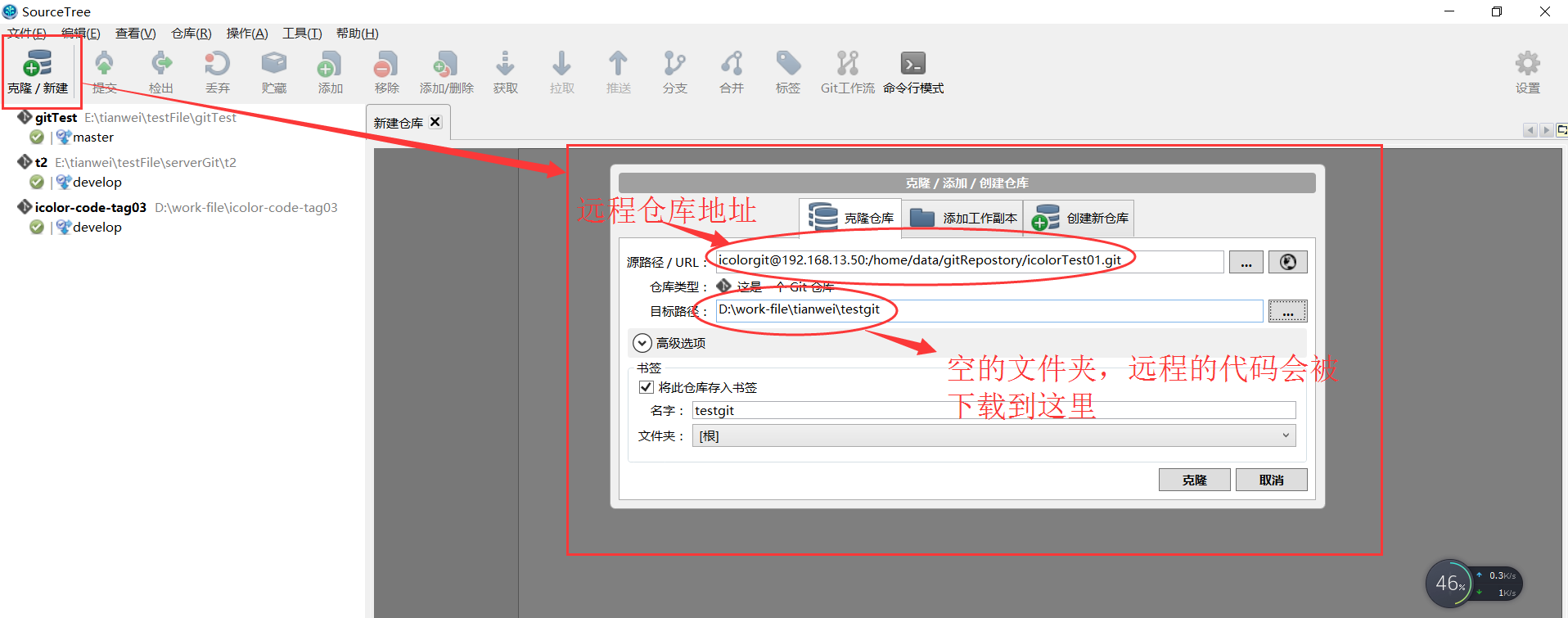
我用使用局域网中的远程仓库地址为:

[icolorgit@192.168.13.50:/home/icolorgit/data/gitRepostory/icolor.git](mailto:icolorgit@192.168.13.50:/home/icolorgit/data/gitRepostory/icolor.git)

局域网中进行测试练习使用的远程仓库地址为：

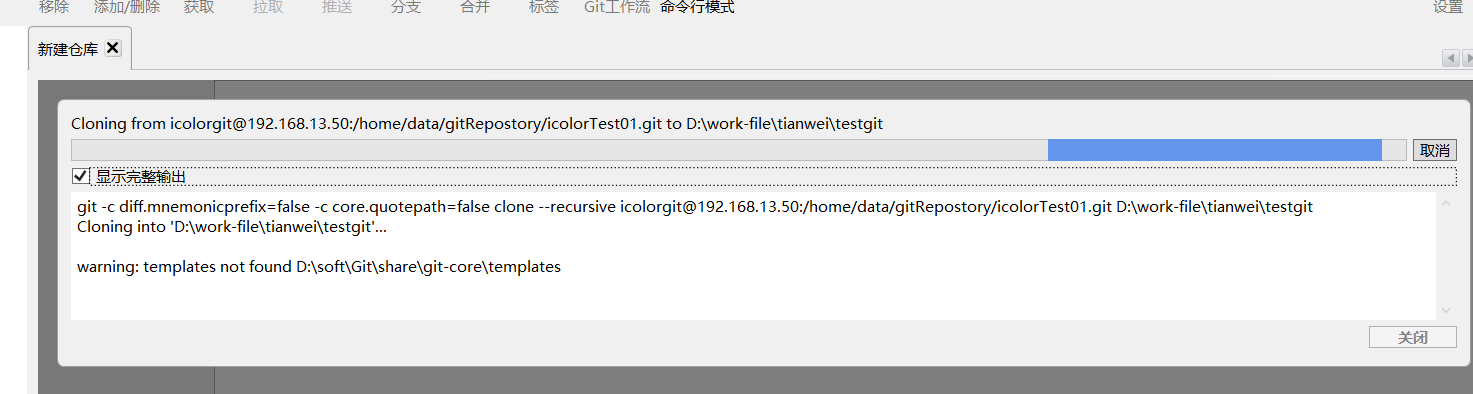
icolorgit@192.168.13.50:/home/data/gitRepostory/icolorTest01.git

sourceTree中进行远程仓库克隆：

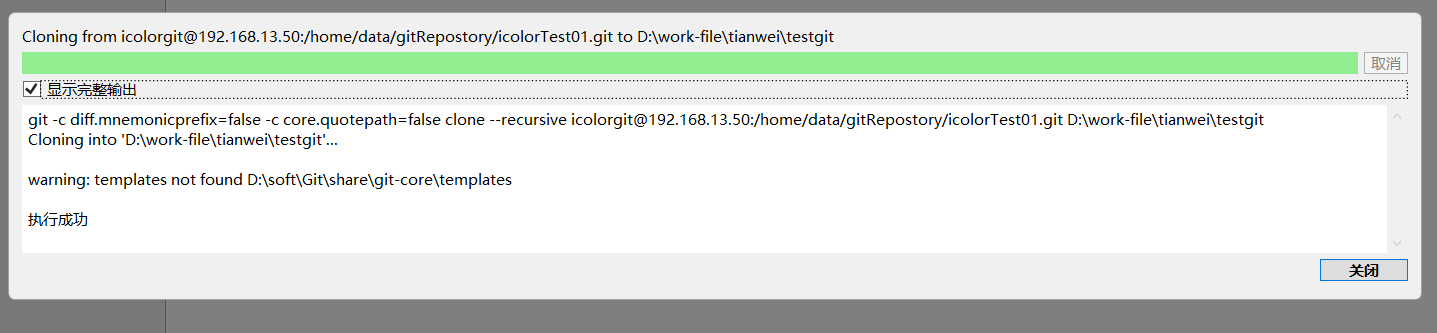


如图所示，可将远程代码下载到本地。

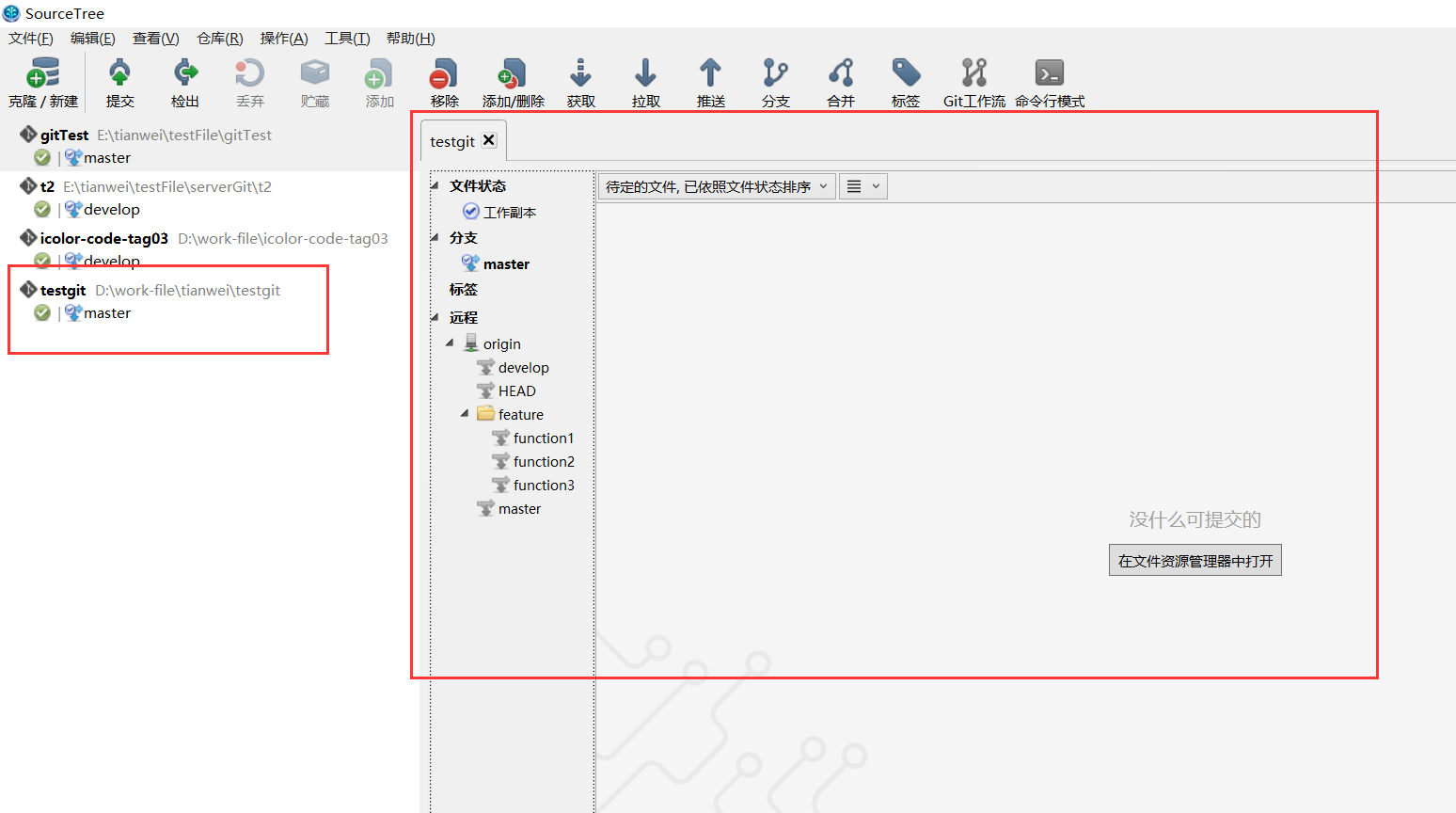
代码下载过程如图：



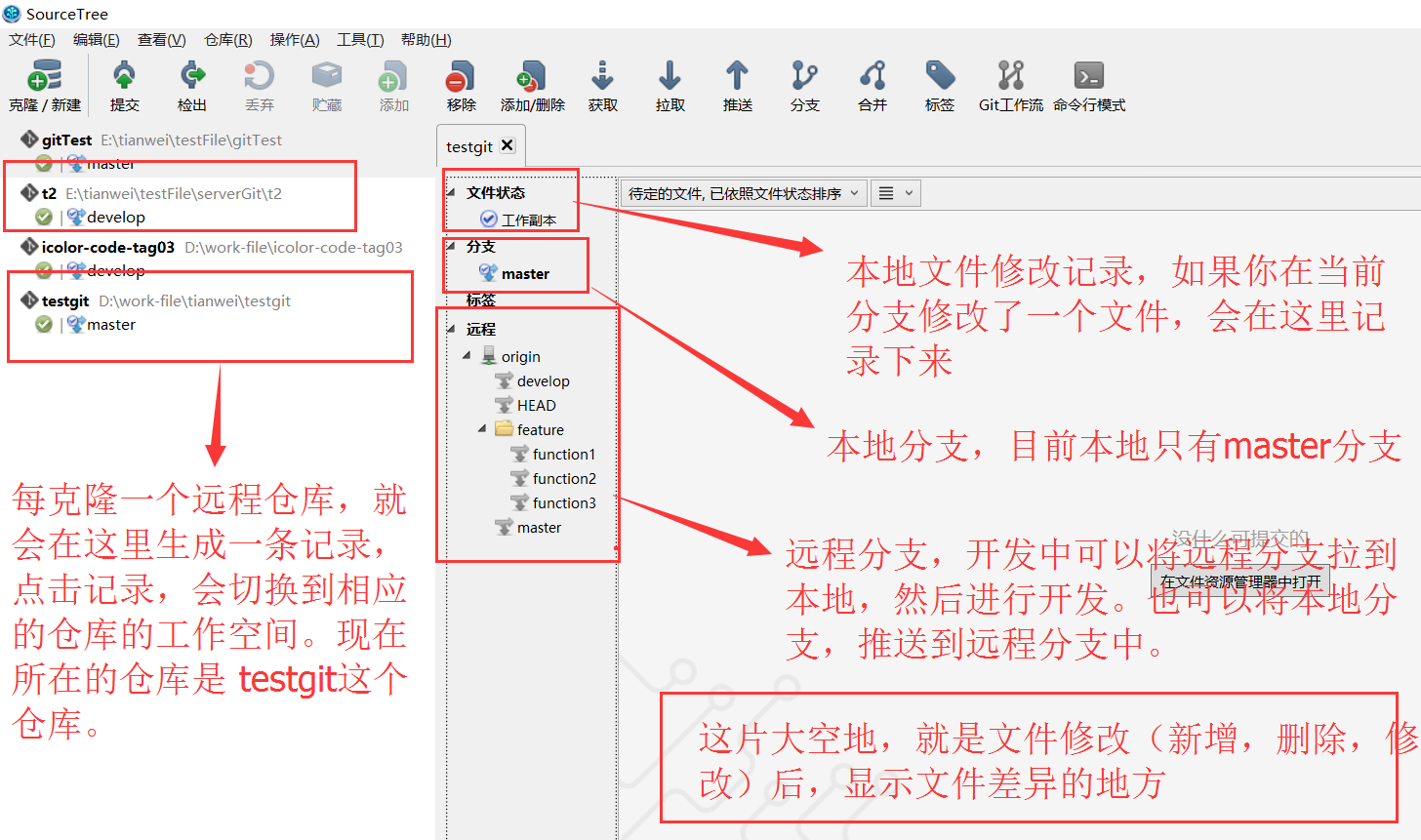
下载完成之后：



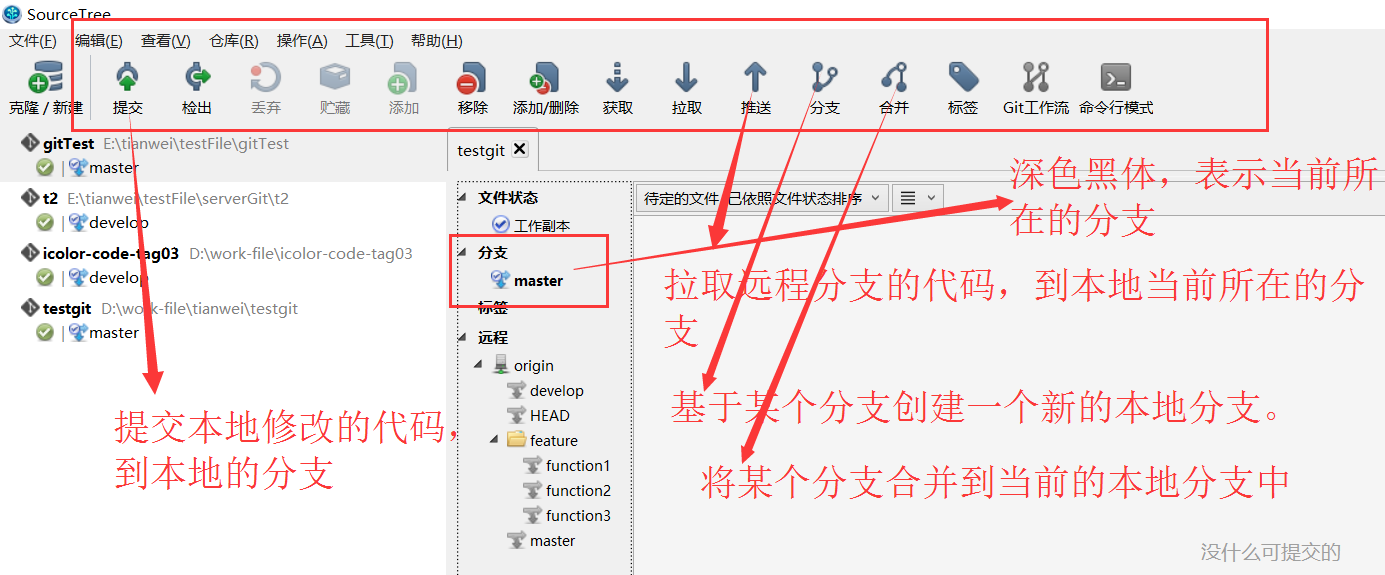
左边工作空间显示如下：



* 1. sourceTree功能说明



* 1. sourceTree目录说明

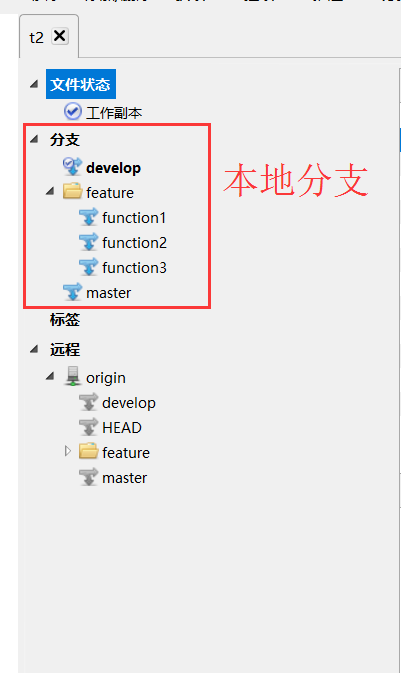


以上只列出我们工作中，所必须了解的的菜单功能，其它功能请自行研究。

* 1. 使用git进行项目开发

我们可以认为git中是以任务的方式进行提交代码的。我们把修改一个bug，增加一个功能，都做为单独的任务进行。每个独立的任务都需要去建立一个自己的分支，在自己的分支上完成后，合并到指定的分支上。

项目开发中，我们是基于develop分支进行开发。克隆下来代码之后，只有master分支，我们会基于master分支创建一个develop分支。开发人员，在develop分支上，新建自己的分支进行开发。



黑色加粗的develop分支，就是当前我们所在的本地分支。这个分支，是基于本地的master分支新建出来的。

我们在进行开发时，如果没有特定说明，就基于develop分支新建自己的分支，进行开发。如果有特殊说明，比如，上线的某个版本有问题，需要及时修改，这时候上线的可能不是develop，我们需要确认在哪个分支上进行修改。确认分支之后，基于该分支，新建自己的hotfix分支进行修复。

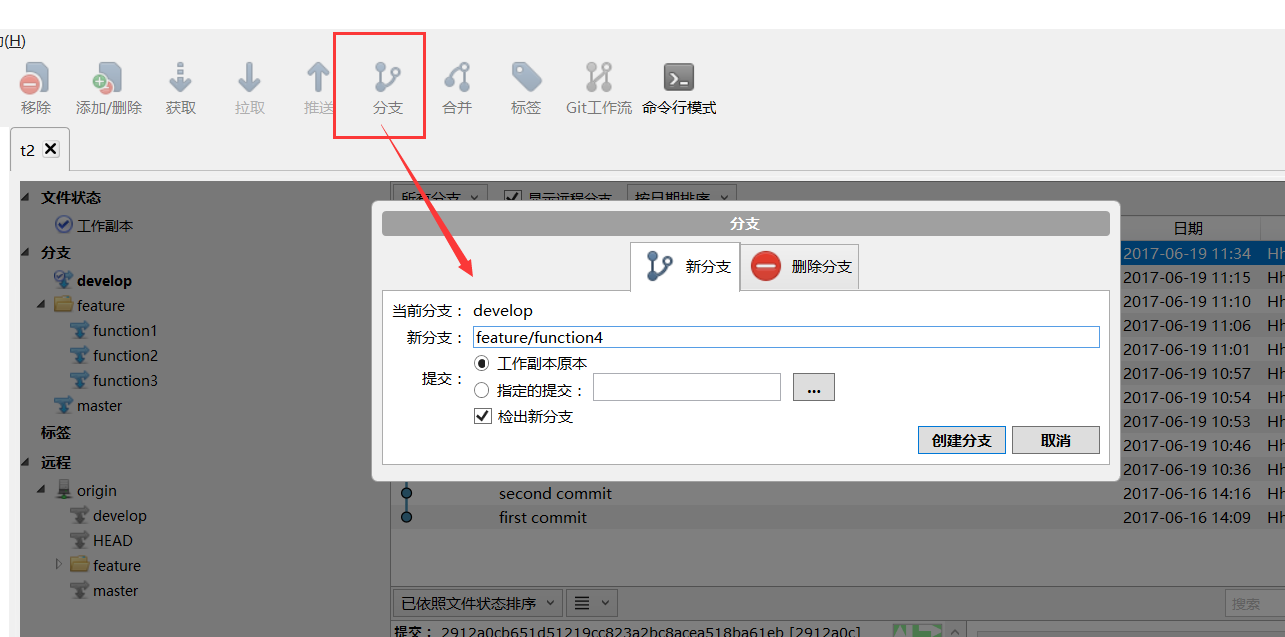
* + 1. 以示例的方式进行sourceTree使用说明

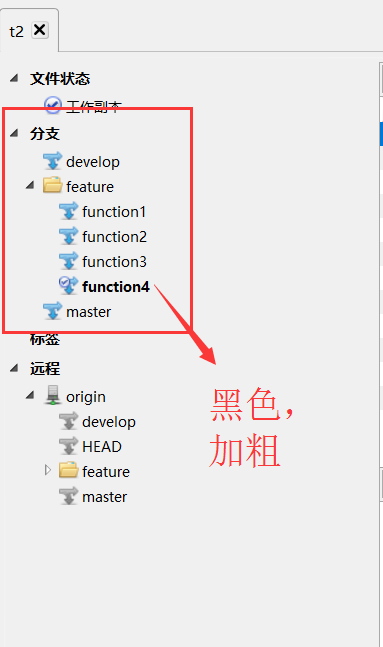
任务1: 在项目中，新增加一个c.txt文件。

操作：

没有说明在哪个分支上开发，所以我们默认为develop分支进行开发。

首先，基于develop分支，新建自己的工作分支，分支名为：feature/function4。



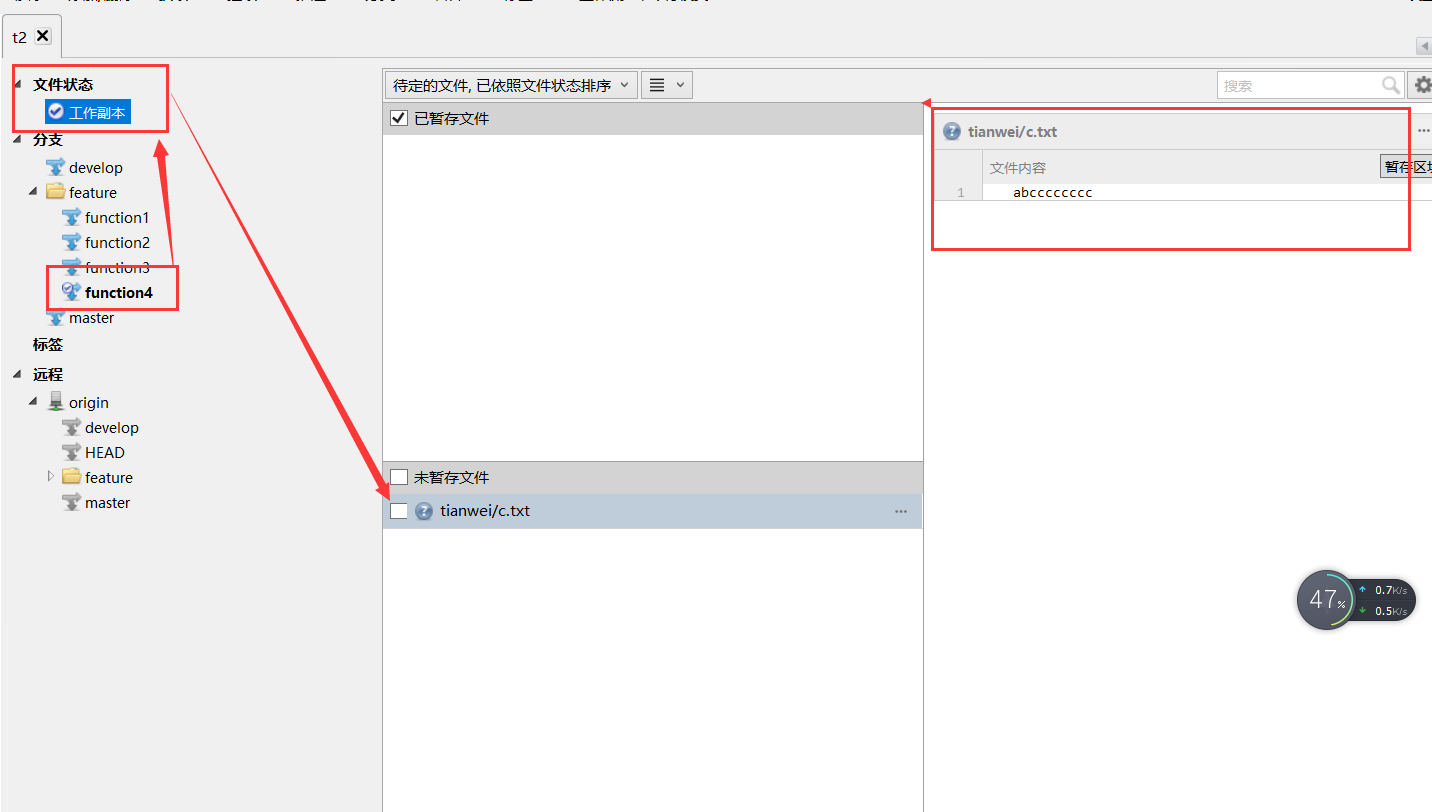


然后我们在当前自己的分支下进行任务：

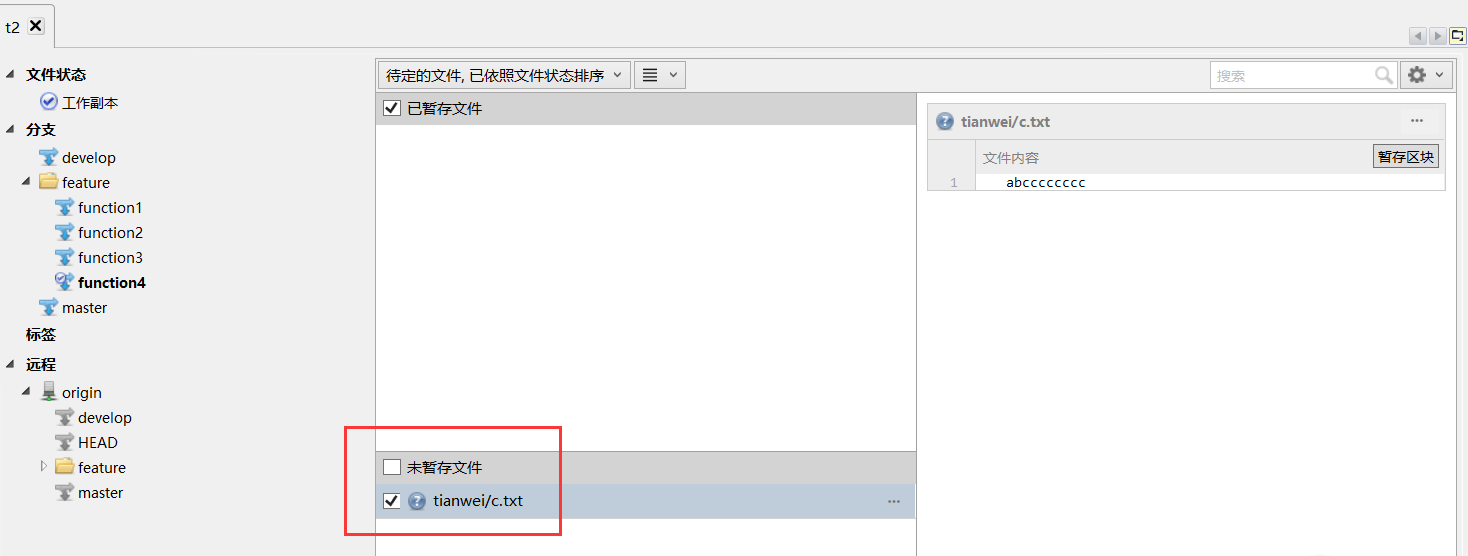


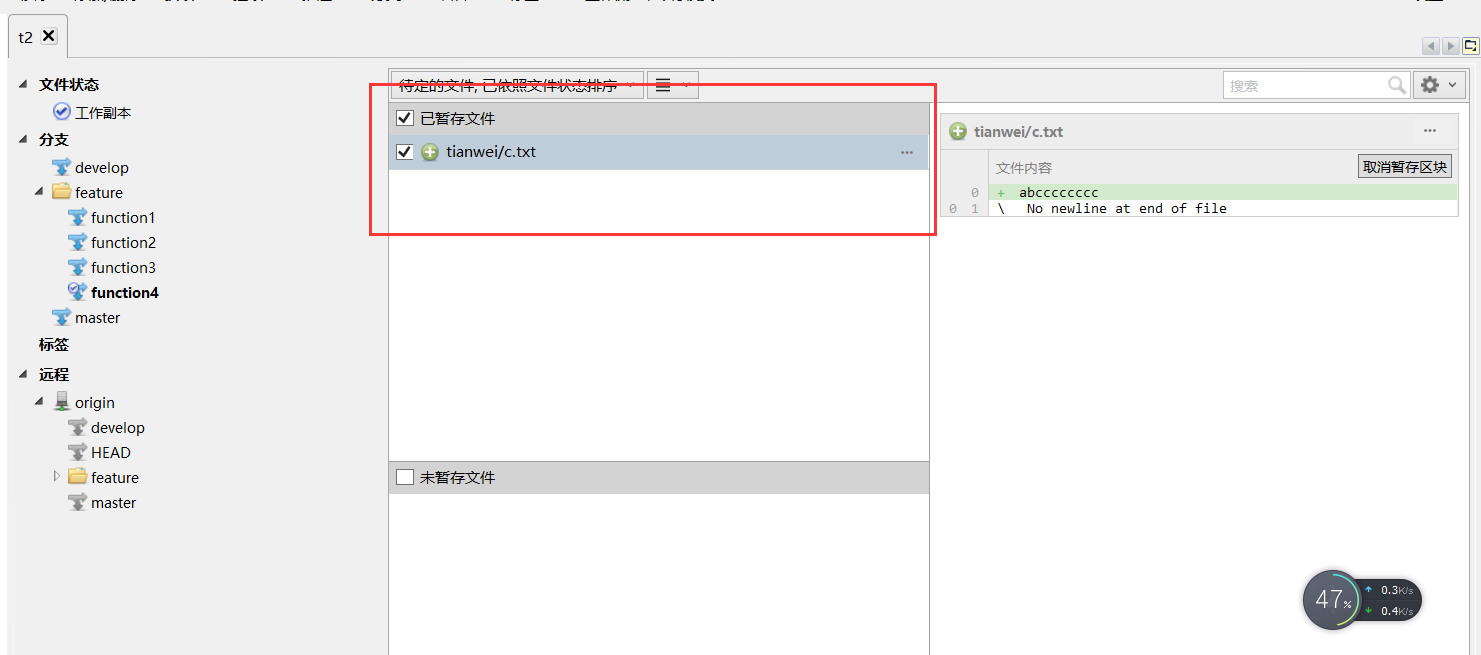
任务完成，开始提交代码：

sourceTree的工作空间中，可以查看到有一个未提交的文件，就是我们完成的任务。

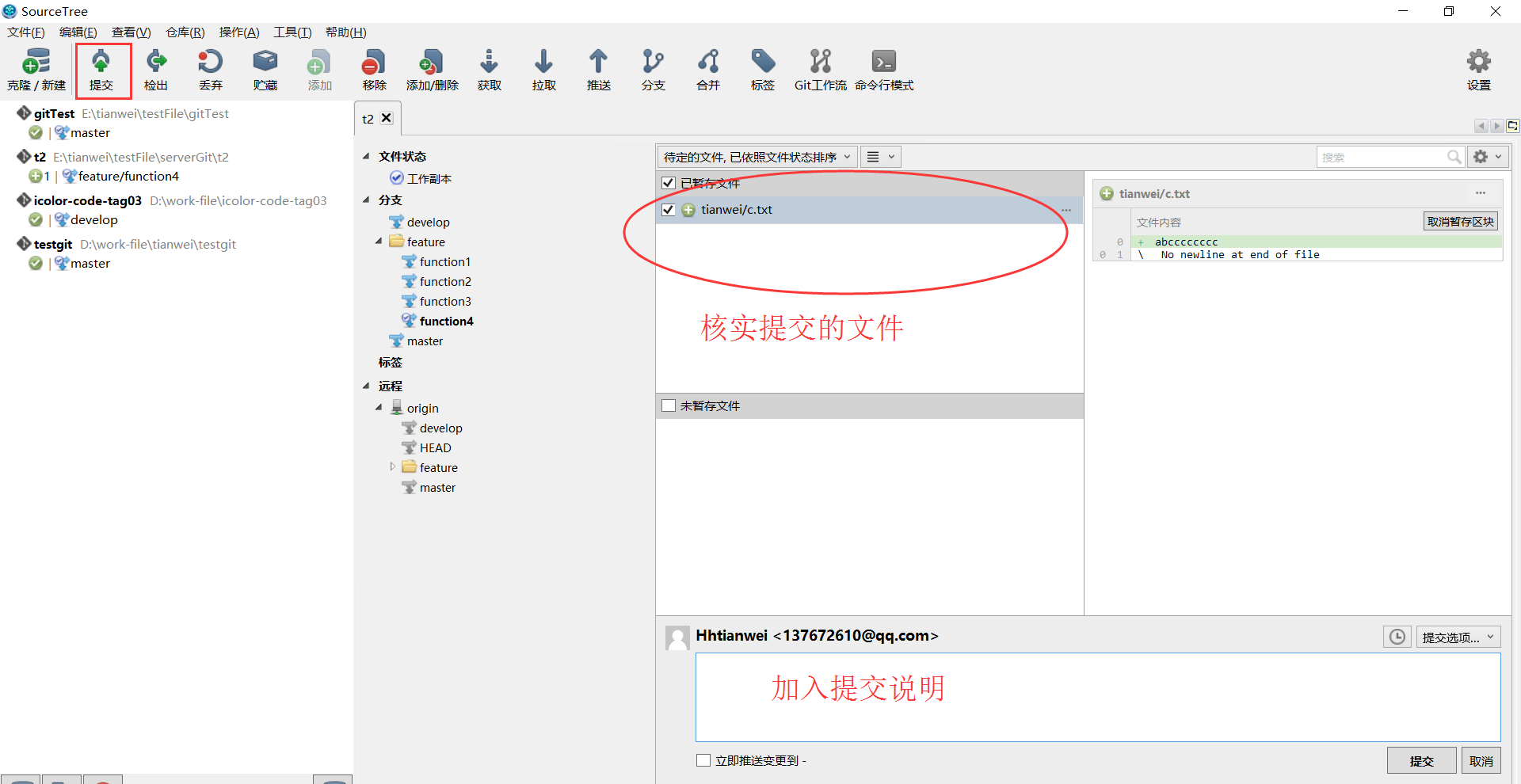


将文件加入到我们的提交队列中：

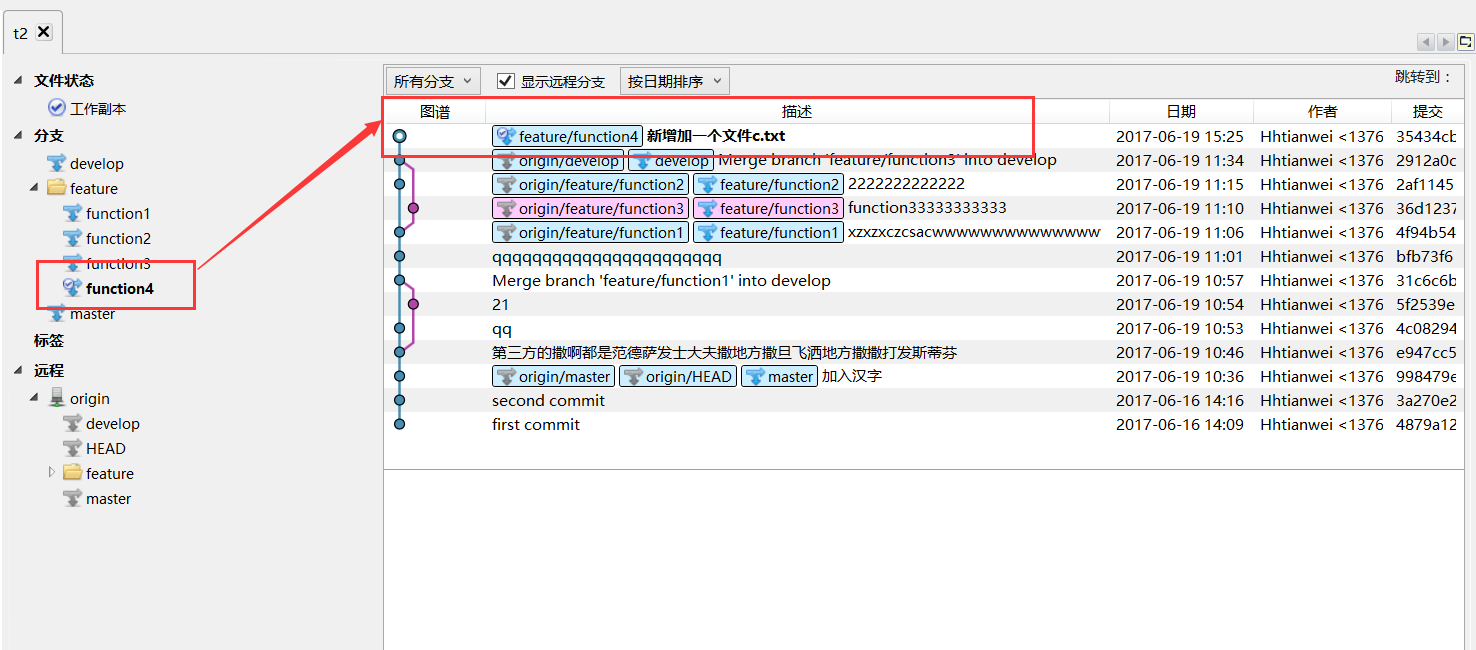




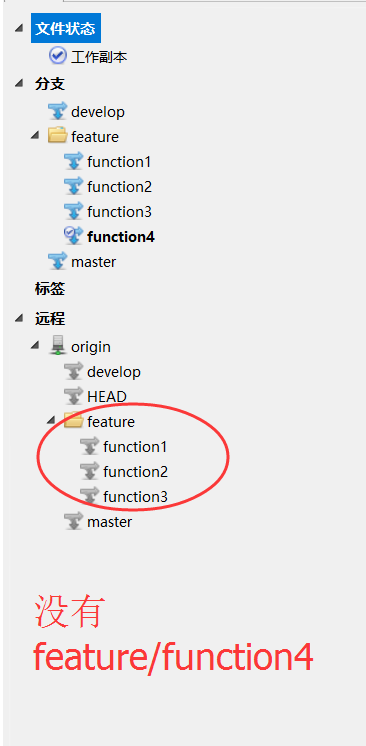
对任务进行提交：



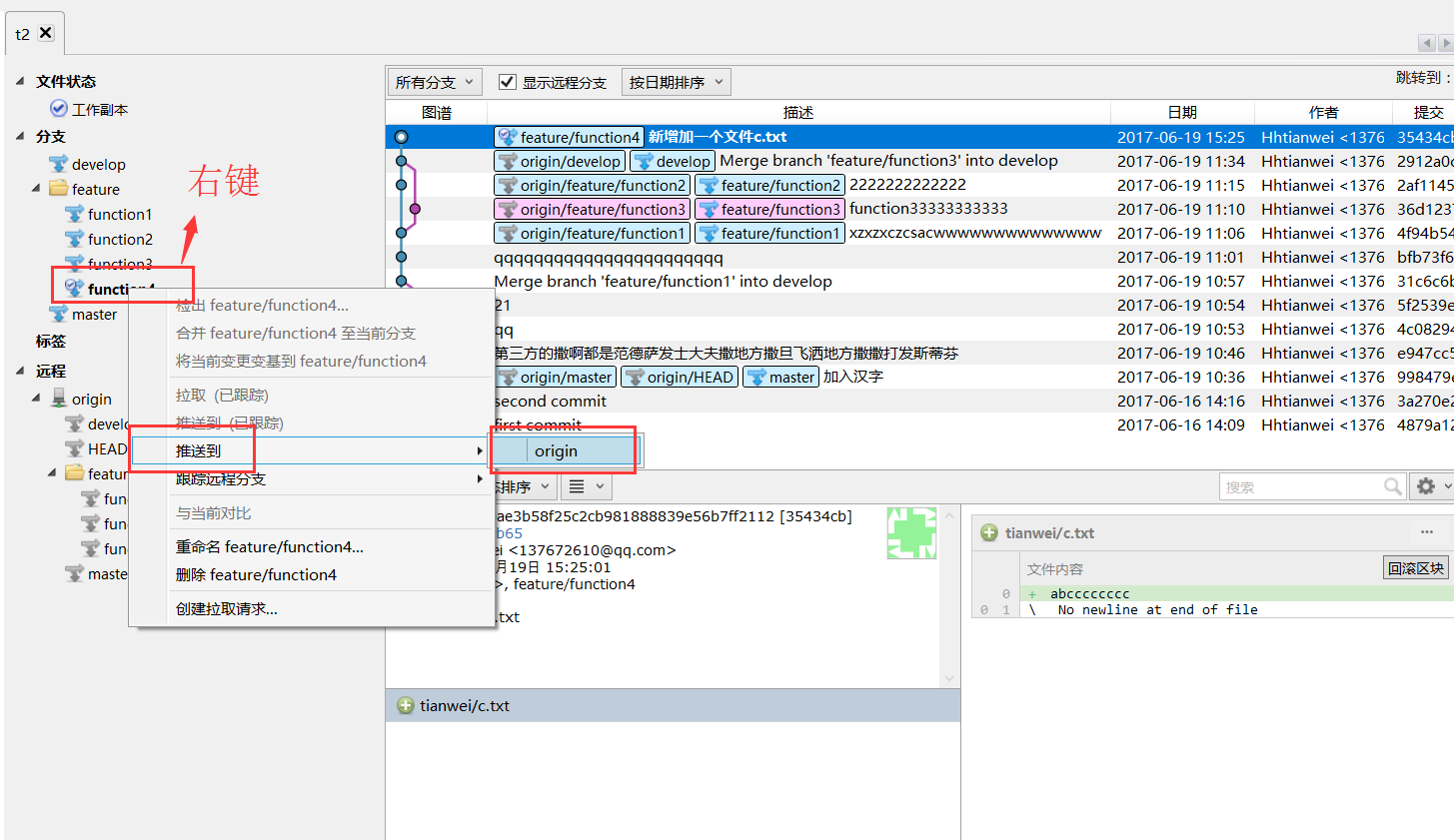
提交之后，会有如下结果：

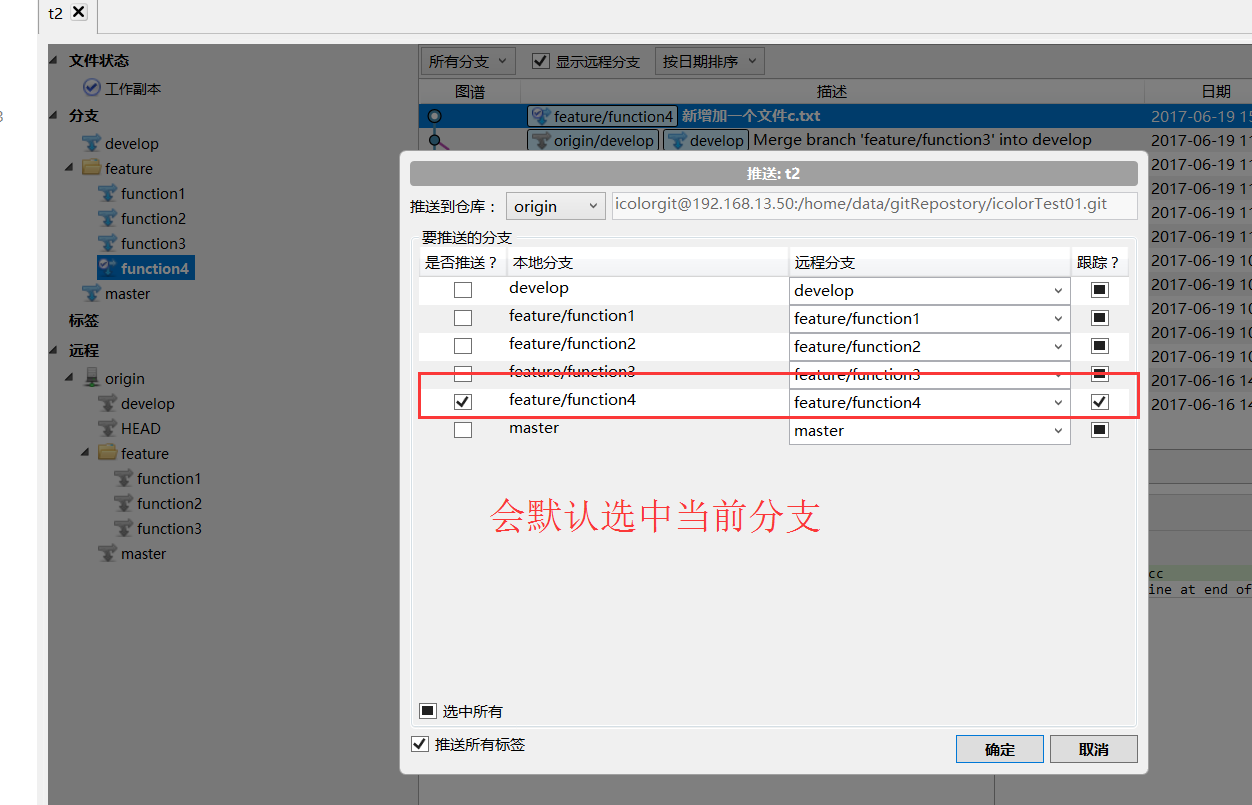


这说明，我们已经把任务提交到本地的分支了。接下来，要把任务，推送到远程的这个分支上。目前远程服务器没有这个分支，我们在推送时，会自动帮我们新建立这个分支。

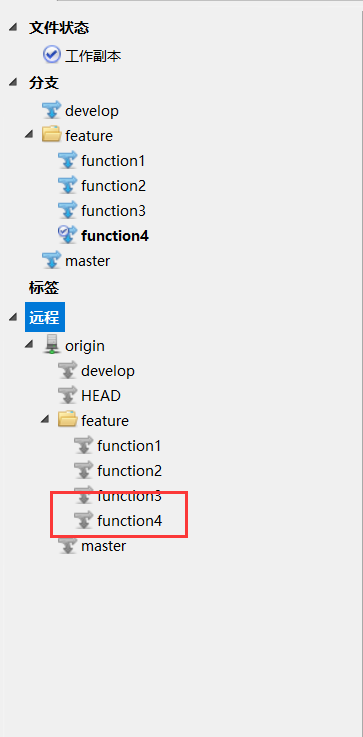


推送本地分支到远程分支：

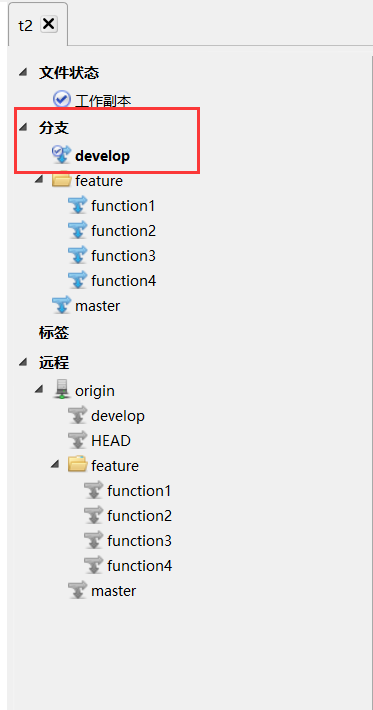




刷新远程分支 F5,发现feature/function4已经出现在远程分支。

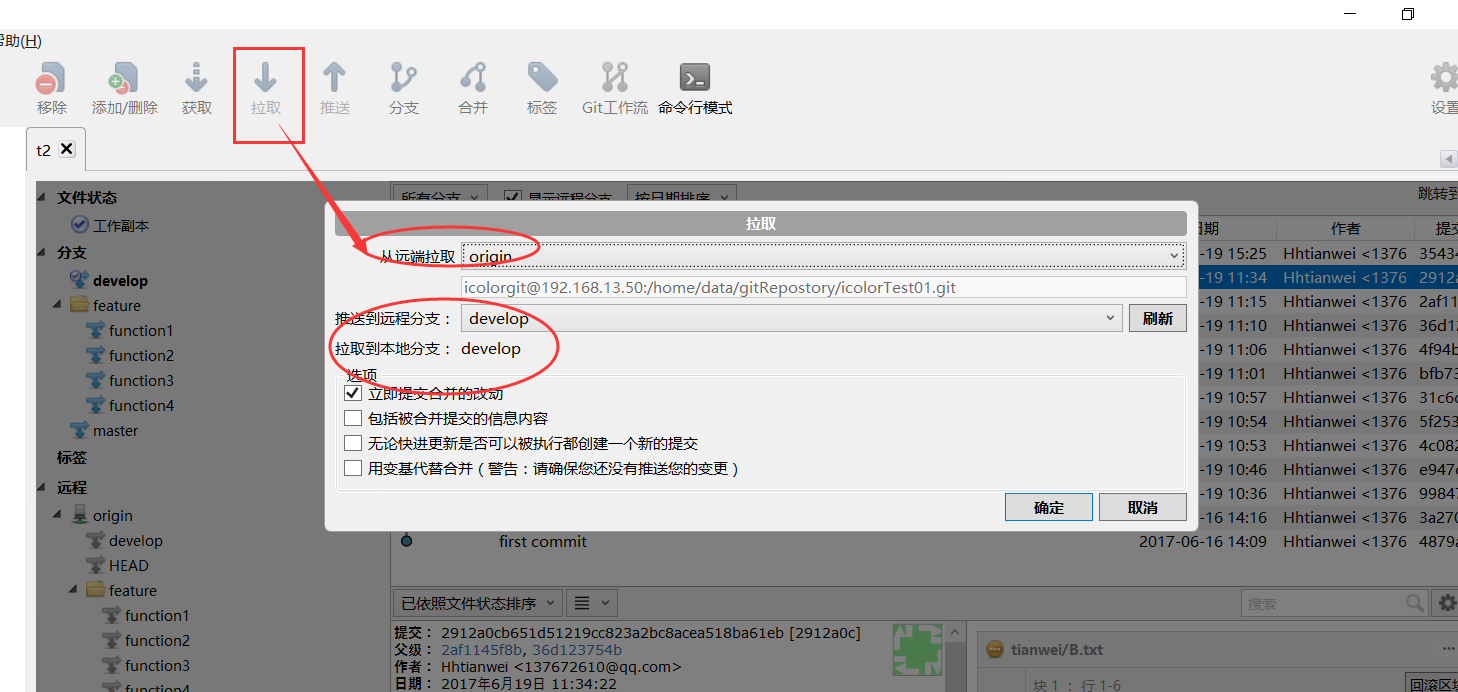


接下来，我们把feature/function4这个分支合并到develop分支上。首先，需要切换到本地的develop分支上。双击本地的develop分支，等待几秒钟，查看结果：

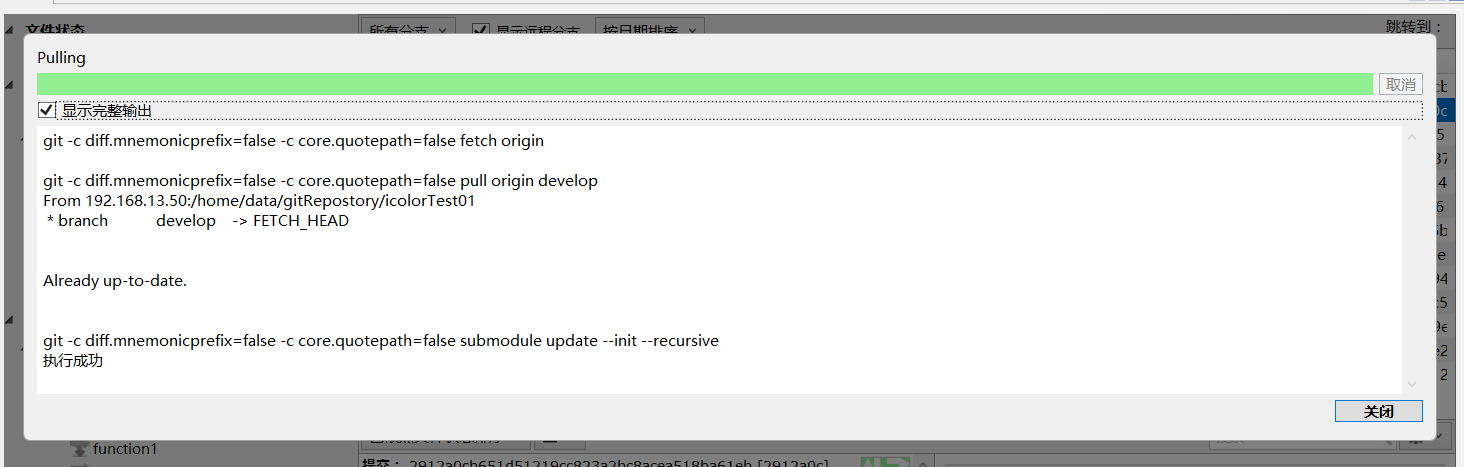


develop分支黑体加粗，说明已经切换过来。

在develop分支上，我们先更新develop代码。也就是说，把develop远程分支的代码拉到本地的develop分支上。因为在我们进行开发的这个时间段，可能有其它人对远程develop分支进行过修改。

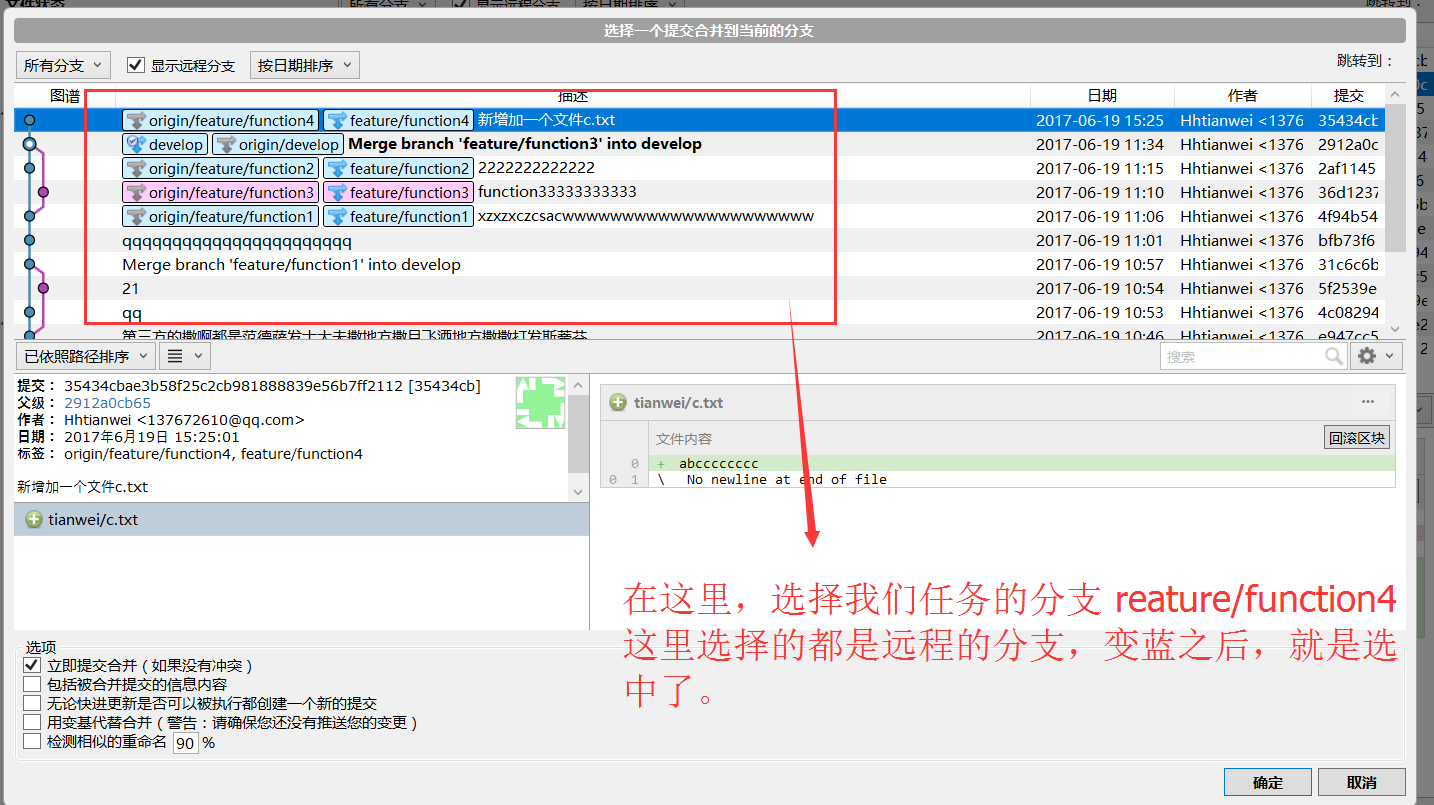


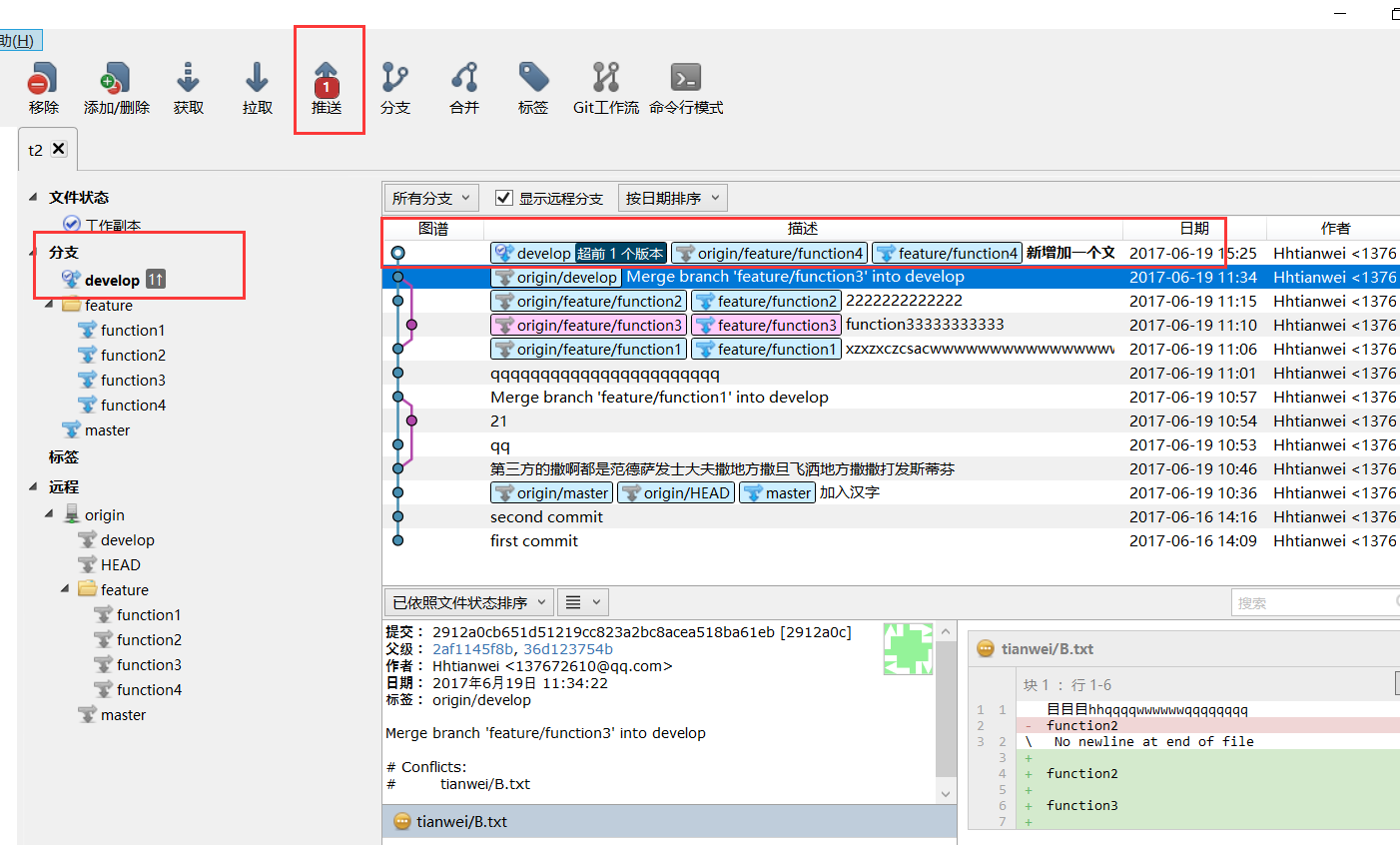
执行结果如下：



以时，如果没有人在我们拉取代码后，再次更新了远程develop分支，那么，我们现在本地的develop分支就是最新的代码。

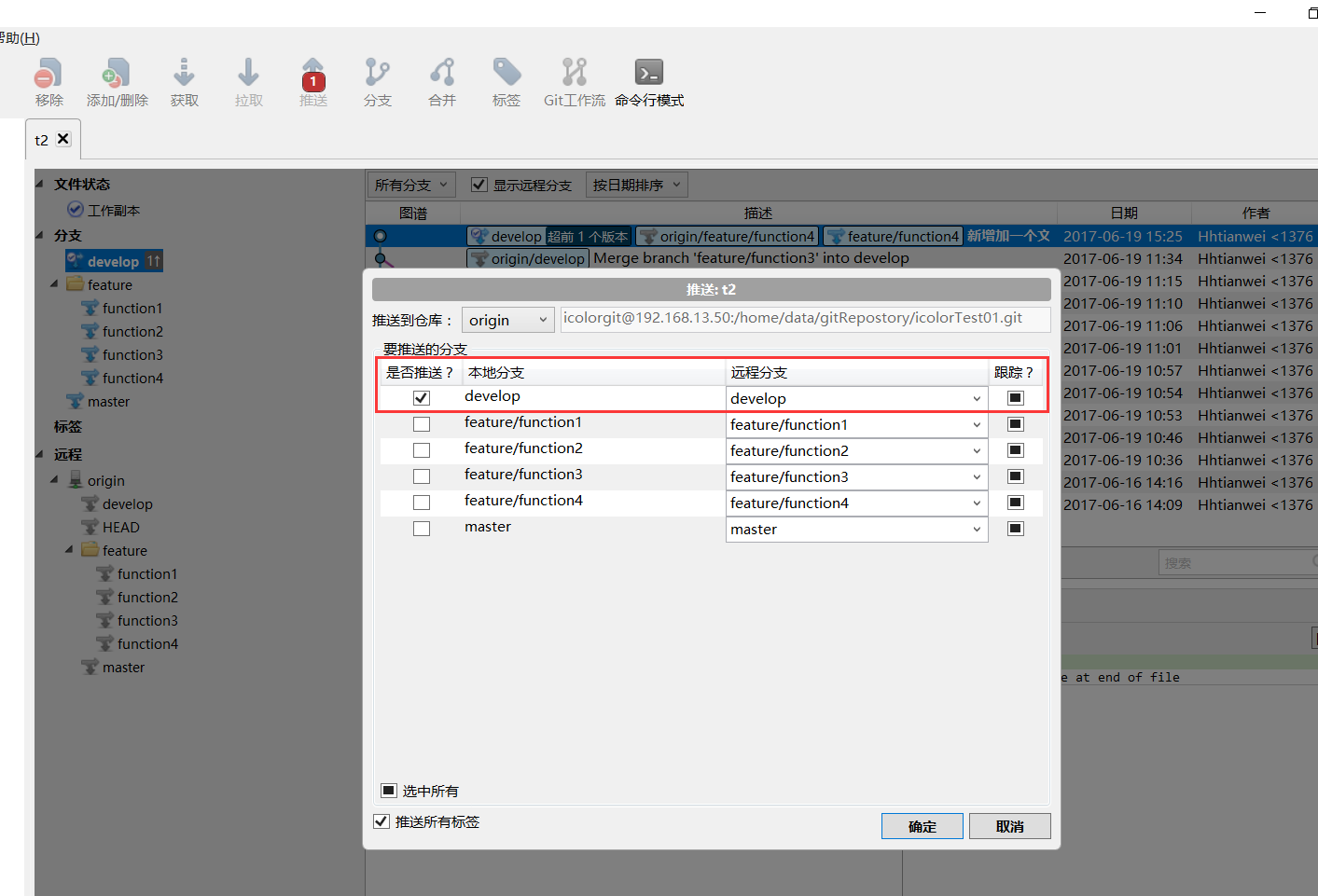
然后把我们的任务feature/function4合并到本地的develop上。然后再把develop推送到远程分支，这样就完成了我们任务到develop的过程。



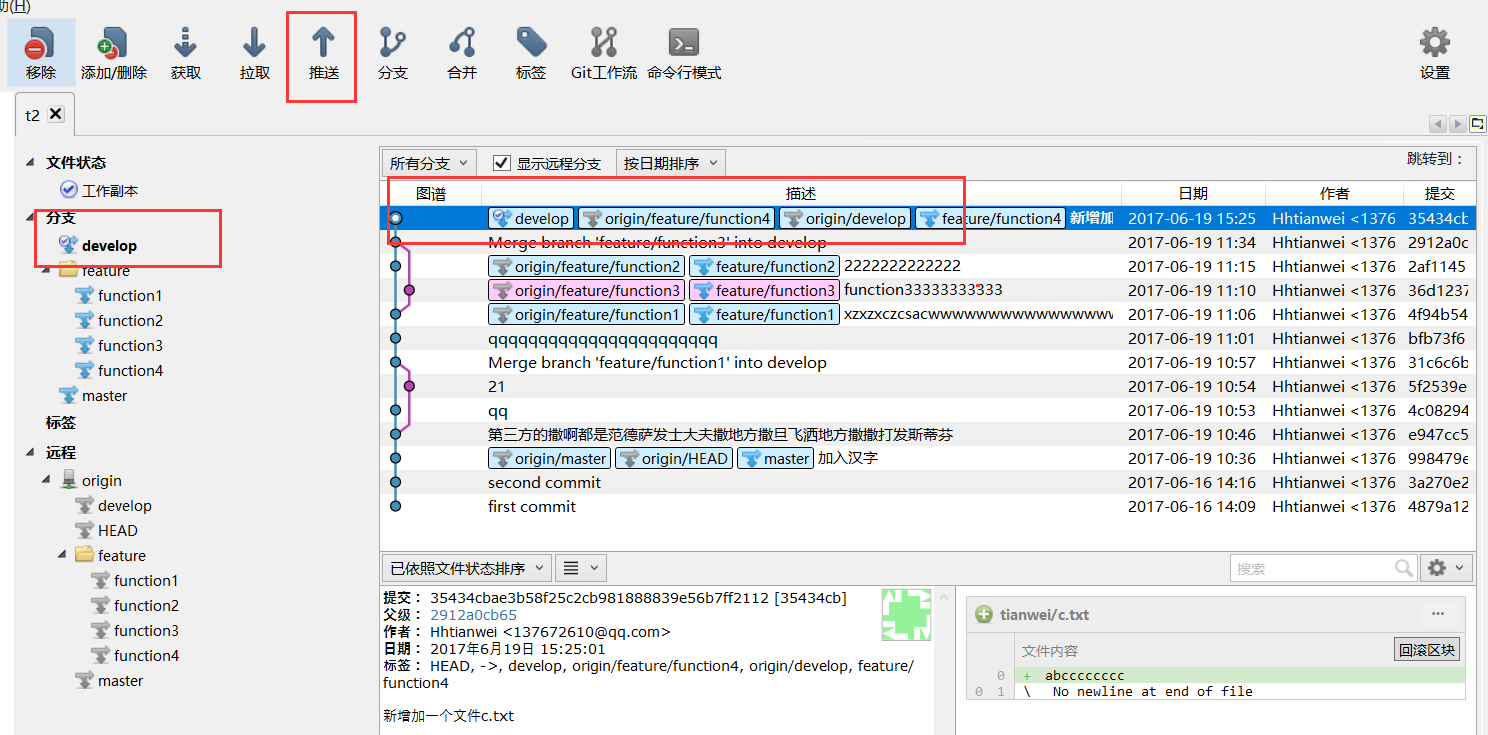
合并之后，出现以上截图。带有数字 1 和向上箭头的部分说明，当前本地分支，与远程分支的差距，也就是说 本地有一个文件没有推送到远程分支上。

现在本地develop分支已经有我们完成的任务，但是远程develop还是没有。所以， 我们需要把本地的develop推送到远程的develop分支中。

点击推送功能，把本地分支推送到远程分支中，请一定要注意，你选择的分支是否正确：



推送完成之后，如下：



到此为止，我们就把任务成功的完成了。如果有其它人，需要进行开发，他需要先把本地的develop分支更新。然后再基于本地的develop分支，新建自己的分支进行开发。不论，新增文件，修改文件，删除文件，都是这样子操作。