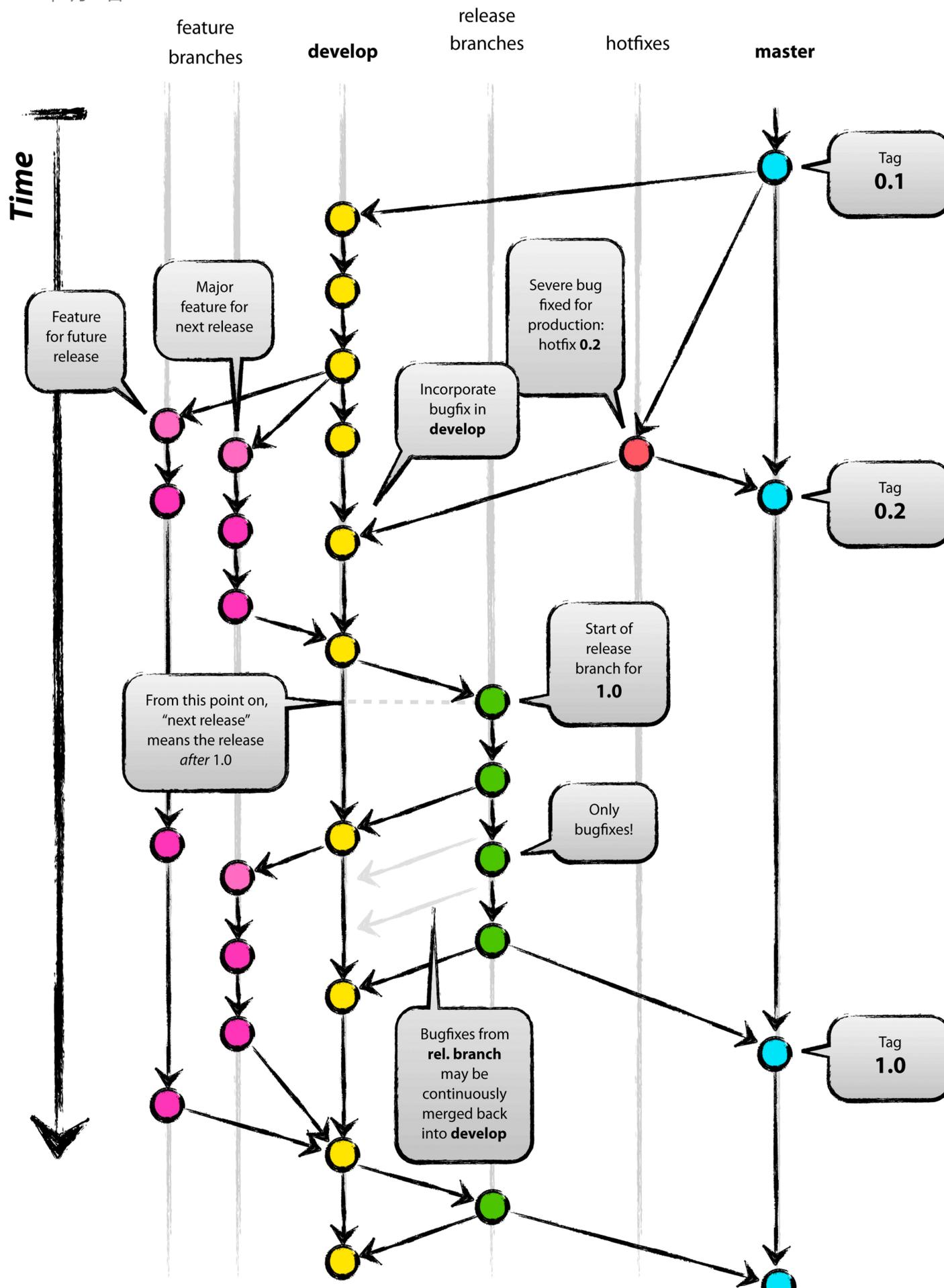


# GitFlow Sharing

2019年3月10日

14:51



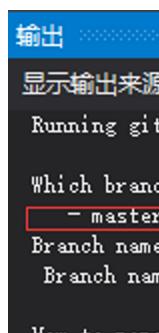
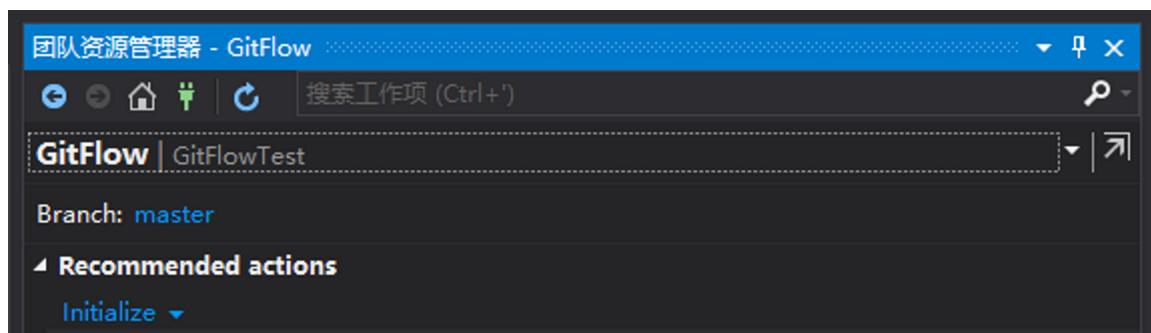
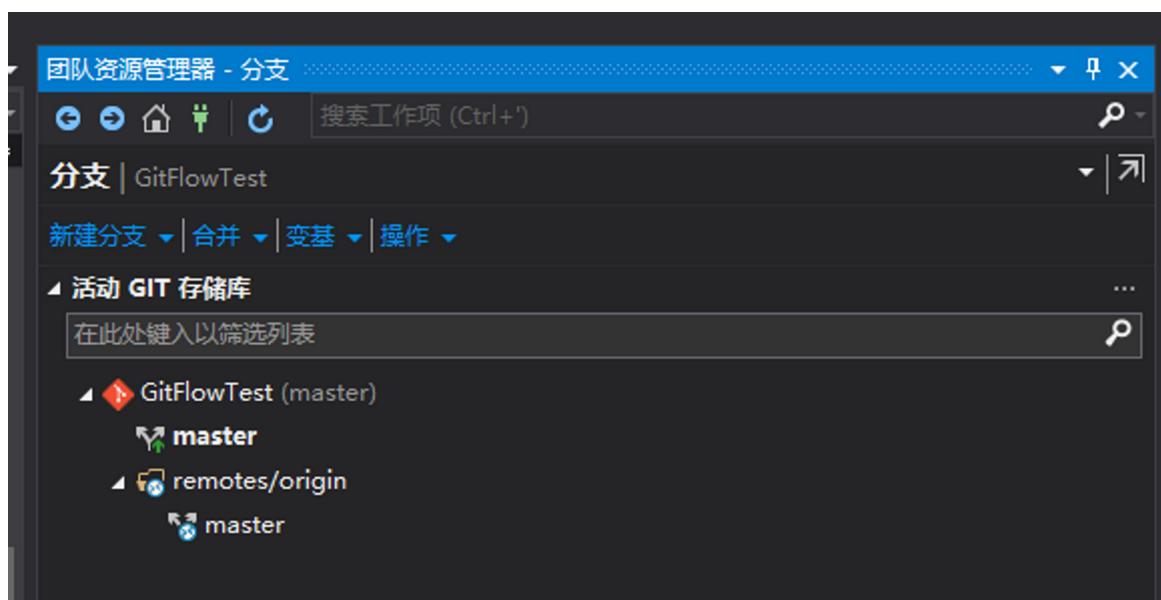
## 1. Why we need GitFlow

### 版本管理的挑战

虽然有这么优秀的版本管理工具，但是我们面对版本管理的时候，依然有非常大得挑战，我们都知道大家工作在

1. 如何开始一个Feature的开发，而不影响别的Feature？
2. 由于很容易创建新分支，分支多了如何管理，时间久了，如何知道每个分支是干什么的？
3. 哪些分支已经合并回了主干？
4. 如何进行Release的管理？开始一个Release的时候如何冻结Feature，如何在Prepare Release的时候，开发
5. 线上代码出Bug了，如何快速修复？而且修复的代码要包含到开发人员的分支以及下一个Release？

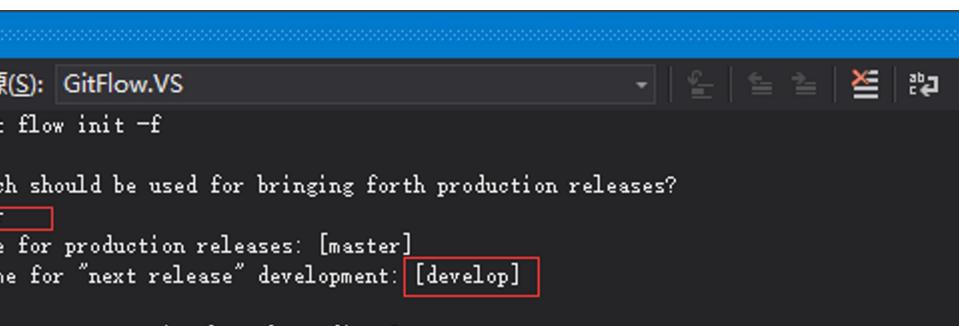
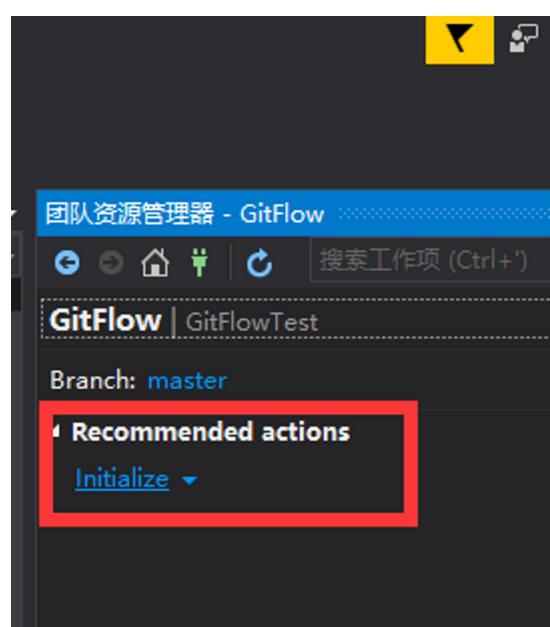
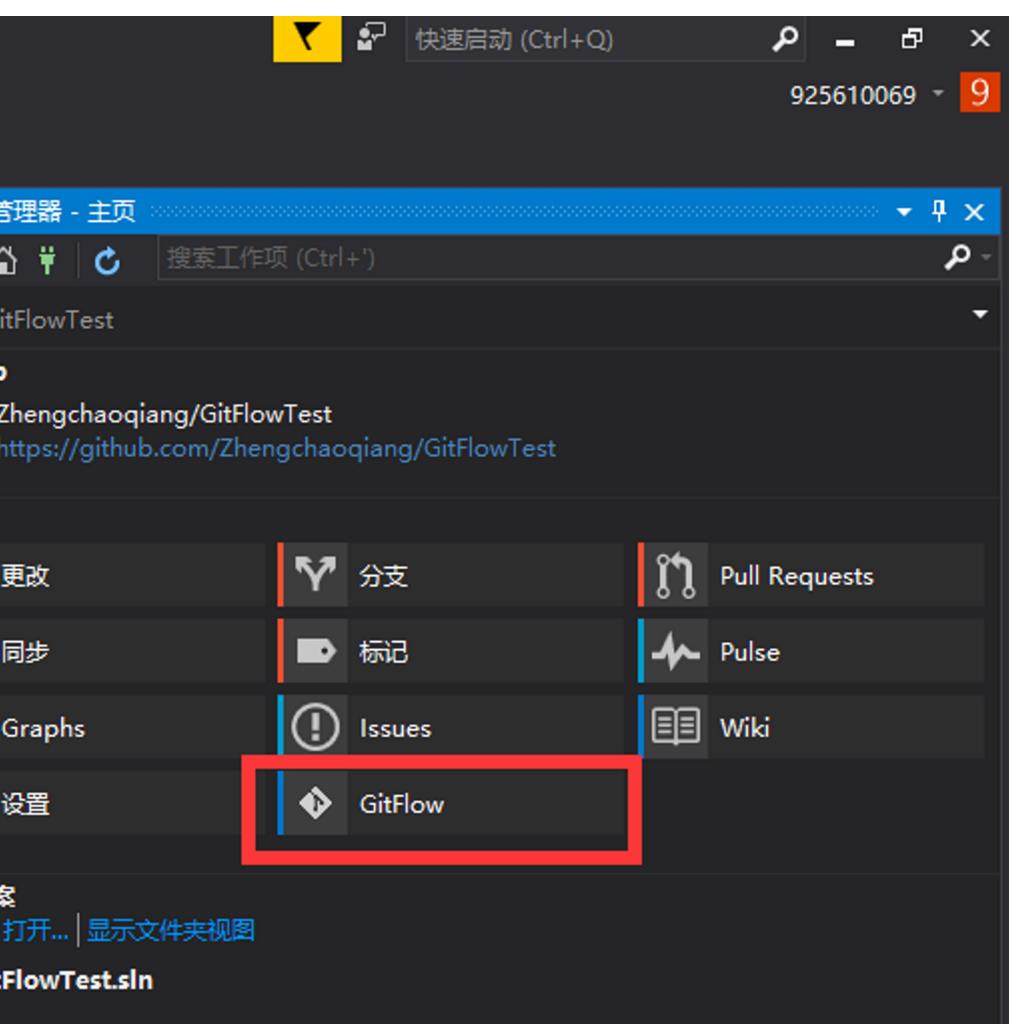
大部分开发人员现在使用Git就只是用三个甚至两个分支，一个是Master，一个是Develop，还有一个是基于Develop的Release，但是人员一多，而且项目周期一长就会出现各种问题。

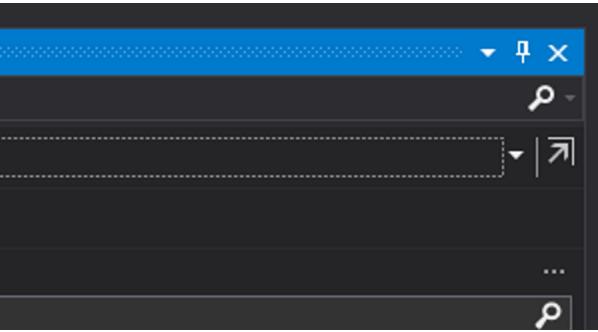
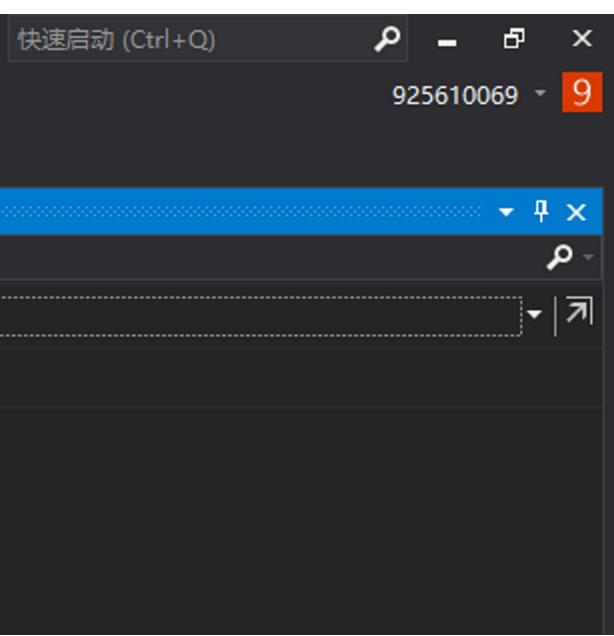


在同一个仓库上，那么彼此的代码协作必然带来很多问题和挑战，如下：

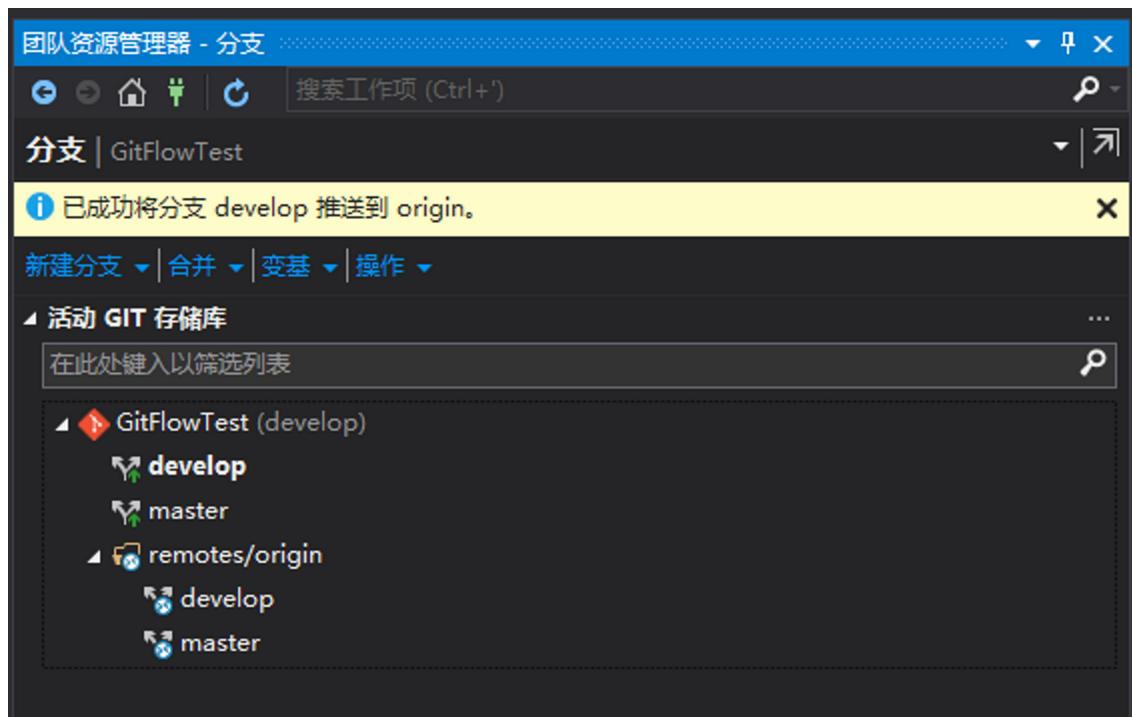
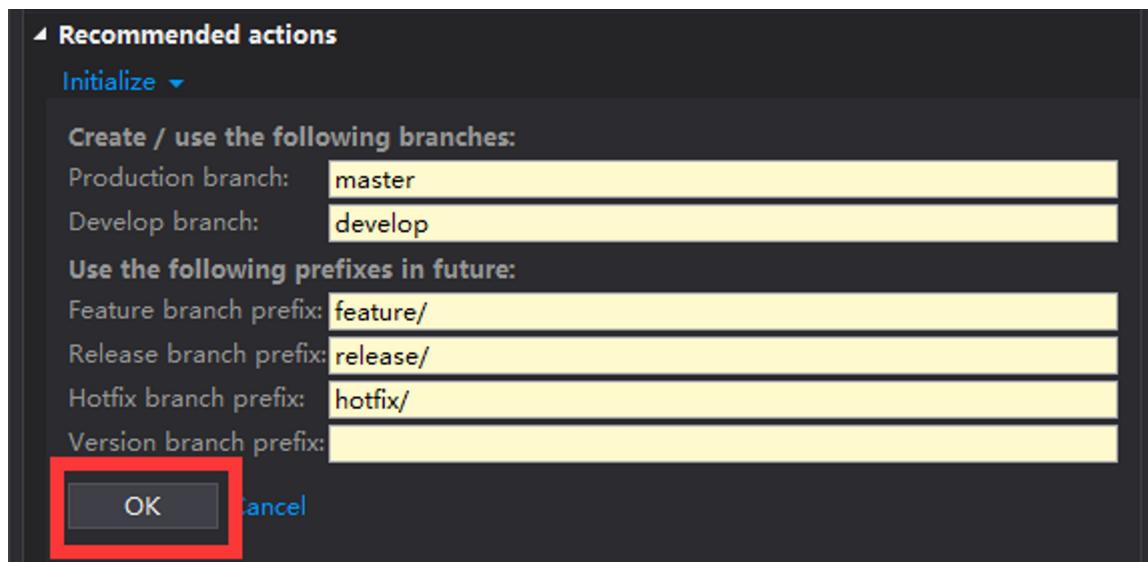
人员可以继续开发新的功能？

lop打得各种分支。这个在小项目规模的时候还勉强可以支撑，因为很多人做项目就只有一个



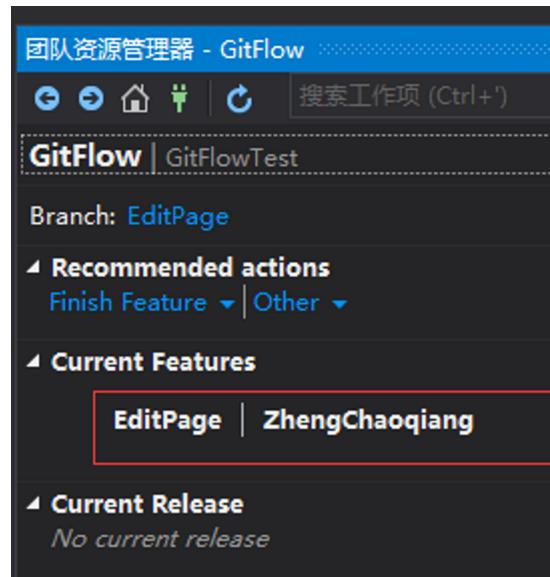
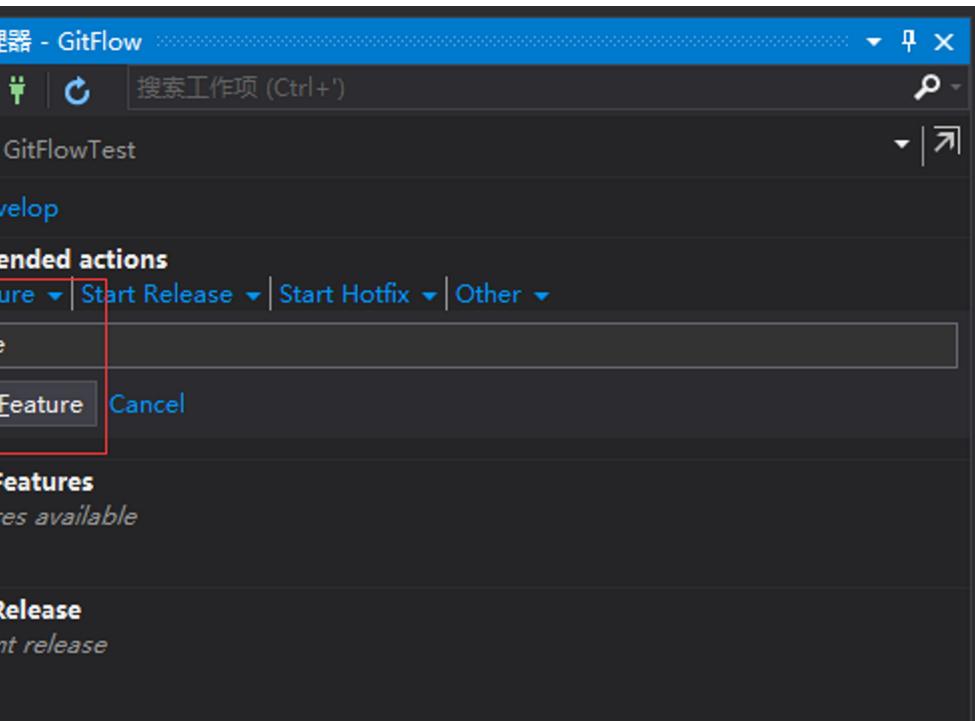


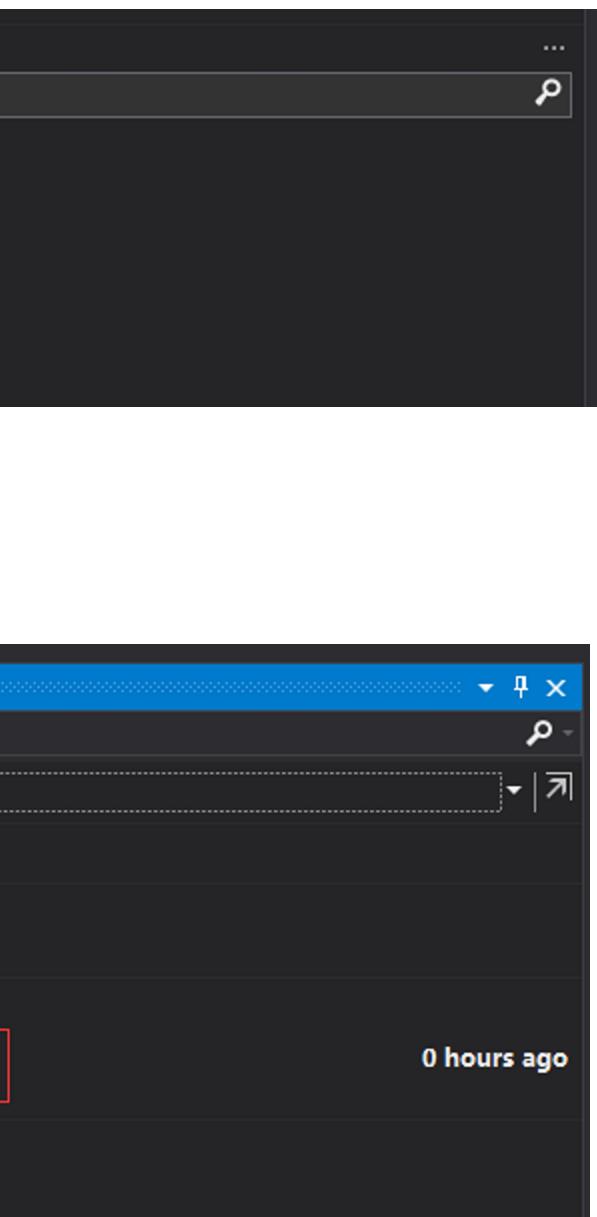




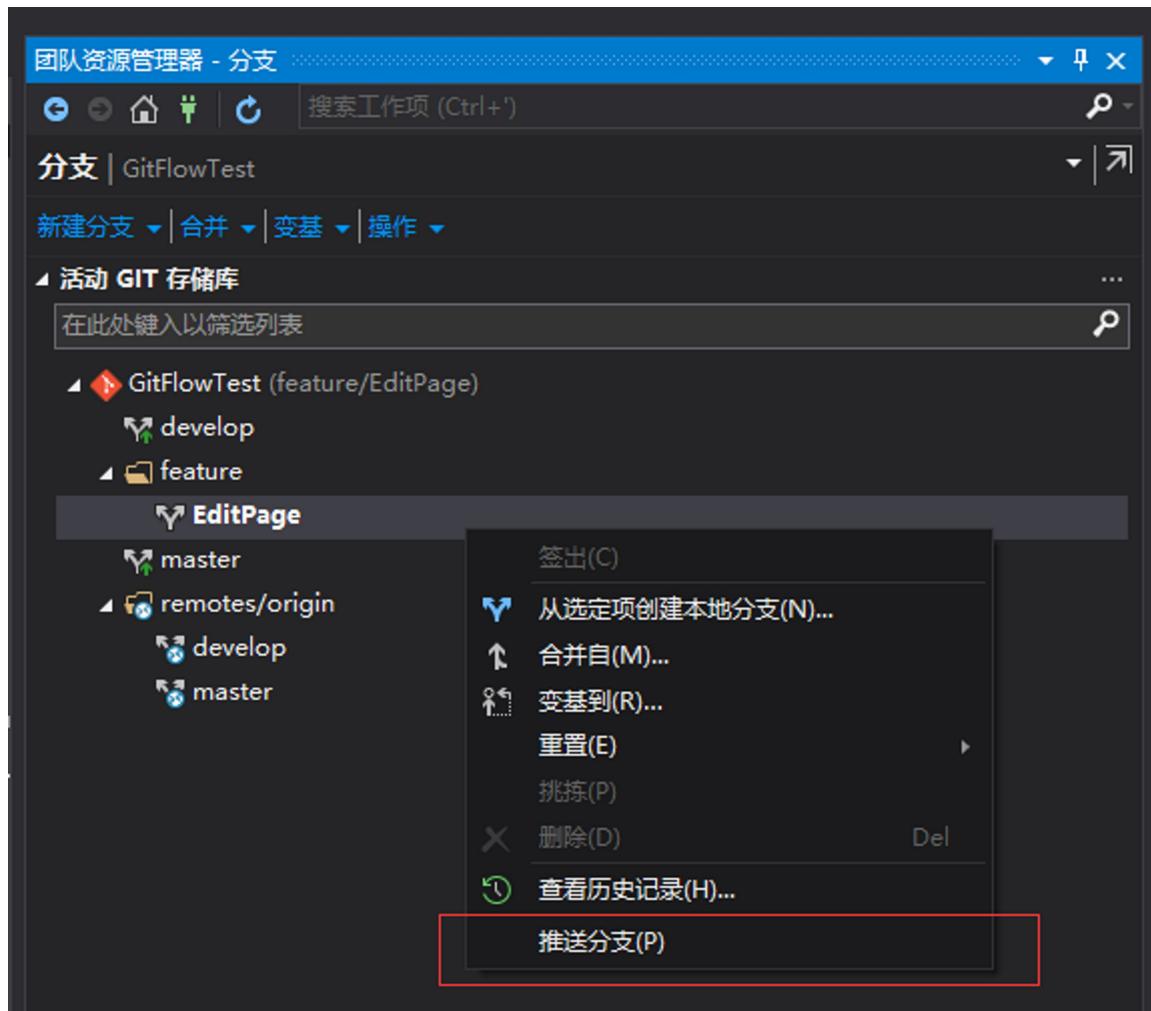
```
for production releases: [master]
for "next release" development: [develop]

your supporting branch prefixes?
branches? []
branches? []
branches? []
branches? []
branches? []
branches? []
tag prefix? []
filters directory? [E:/GitHub/GitFlow/HelloWorld/GitFlowTest/.git/hooks]
```









```
显示输出来源(S): GitFlow.VS
```

```
Running git flow feature finish "EditPage"
Your branch is up to date with 'origin/develop'.
Updating be6a043..b29cb4a
Fast-forward
  WinFormAppTest/Form1.cs | 1 +
  1 file changed, 1 insertion(+)
The command is taking longer than expected
Deleted branch feature/EditPage (was b29cb4a).

Summary of actions:
- The feature branch 'feature/EditPage' was merged into 'develop'
- Feature branch 'feature/EditPage' has been locally deleted; it has been remotely deleted from 'origin'
- You are now on branch 'develop'
```

