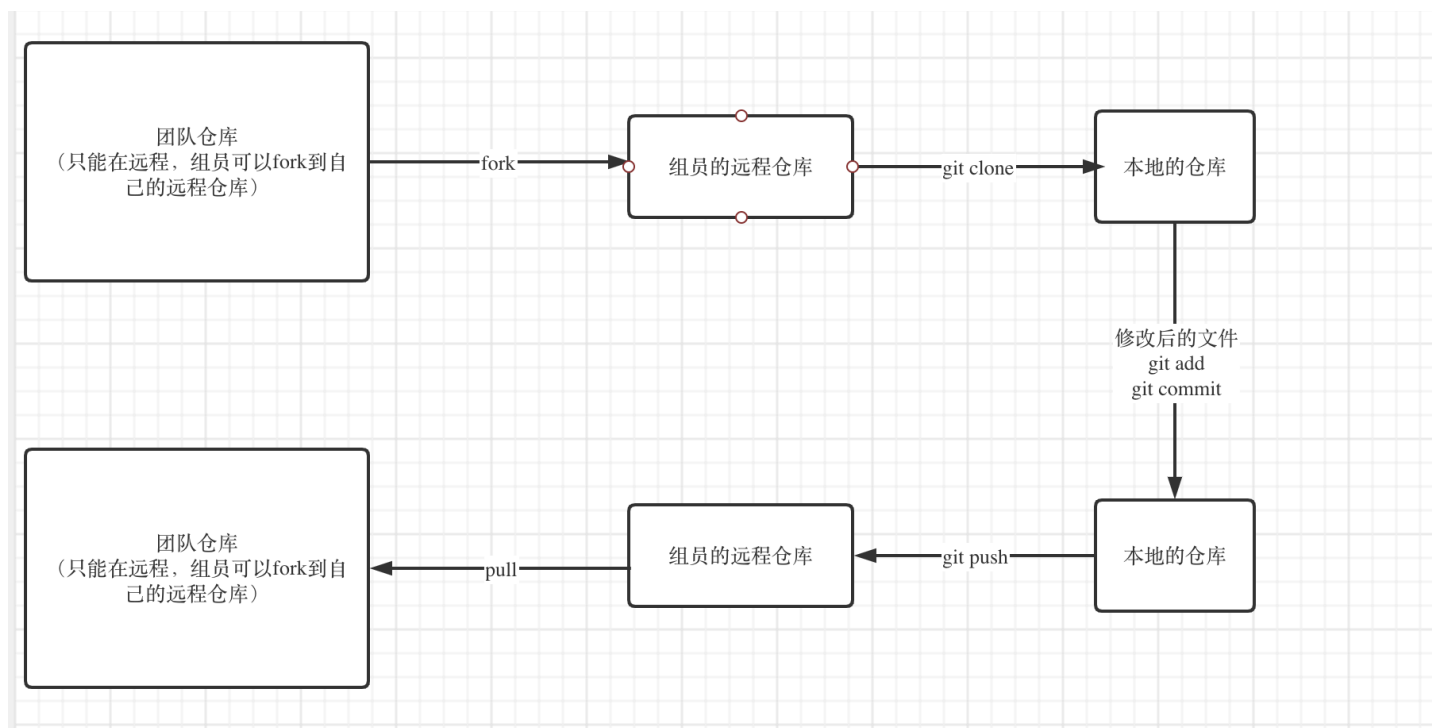


Git团队合作

一、创建开发分支

1. 队长先建个组织，然后创建一个团队的远程仓库(MagicMe)
2. 队长创建dev分支，分支创建完毕后，会自动跳转到dev分支。由于dev分支是从main分支上创建的，因此内容与main分支一致。
3. main分支是稳定版本，一般不轻易改动；dev分支是开发测试用的，可以随便弄。
4. 团队仓库和个人仓库不要混淆，是先从团队仓库fork到自己的远程仓库，有了自己的远程仓库再clone到自己的本地仓库，进行开发修改后，先提交到自己的本地仓库，再提交到自己的远程仓库，再pull到团队仓库。



二、Fork项目到个人的仓库

1. 队员从团队的仓库，fork到自己的远程仓库
2. 然后Clone自己的远程仓库到自己的本地仓库

```
git clone <ssh的地址>
```

3. 此时你自己的本地仓库并没有把自己的远程仓库的dev分支clone下来

```
git branch //查看自己的本地分支，你只能发现只有main分支
git branch -a//查看所有分支，包括你本地仓库的main分支和你的远程仓库的dev分支
git checkout -b dev origin/dev//创建一个本地的dev分支，再把自己远程的仓库的dev分支（origin/dev）的内容放在该分支内，并切换到dev分支。
```

```
git branch//现在就可以查看本地的main分支和dev分支了。
ls 或 dir//可以看到dev分支的内容
```

```
git checkout master
//想切换回master分支的时候，再用 git checkout master
```

三、和团队项目保持同步

//什么时候保持同步呢，每天开始写代码前，都得先保持同步，每天的代码写完后，必须pull。明天重复这个过程。
//换句话说就是，你只要pull了，之后想写代码，得先保持同步。

```
git remote -v//查看有没有设置upstream
//如果没有显示upstream
git remote add upstream 团队项目地址
git remote -v//如果出现upstream，则设置成功
```

```
git fetch upstream//获取团队项目最新版本，此时并没有把最新版本合并到你本地的分支上
git merge upstream/dev//如果当前分支是dev分支，会将源分支（upstream/dev）合并到当前分支（dev）
//如果你是在本地的master分支上开发，那么在使用该命令前，先切换到master分支。
```

//注意此时只是将团队项目的最新版本合并到了本地分支，你的远程仓库并没有合并，所以需要：
git push origin dev//推送本地仓库到远程仓库

四、push修改到自己的项目上

```
//开发修改后
git add .//将所有文件添加到暂存区
git commit -m <message>//提交到自己的本地仓库
git push//提交到自己的远程的仓库
//注意，在当前所在分支使用push，会push到与这个分支相关联的远程仓库分支。这里dev分支与origin/dev关联，因此push到GitHub上的dev分支。
```

五、请求合并到团队项目上

首先到你的GitHub上，进入你的远程仓库里。点击红框处的Pull request

报课系统——团队 — Edit

72 commits 3 branches 0 releases 1 contributor

Your recently pushed branches:

dev (less than a minute ago) [Compare & pull request](#)

Branch: dev [CourseManagement / +](#)

This branch is even with fzu2015:dev. [Pull request](#) [Compare](#)

File	Commit	Time
CourseManagement	Merge pull request #42 from 031302509/master	5 hours ago
docs	Merge pull request #1 from zhengshiyuan/dev	3 days ago
README.md	添加dev分支说明	2 days ago

[Code](#)

[Pull requests](#) 0

[Wiki](#)

[Pulse](#)

[Graphs](#)

[Settings](#)

SSH clone URL

git@github.com:sch [Copy](#)

You can clone with HTTPS,

下图左边红框，表示要合并到fzu2015/CourseManagement项目的dev分支。

下图右边红框，表示要从自己仓库的dev分支发起合并请求。

总的意思就是从自己的远程仓库要合并到团队的仓库

点击红框处的 Create pull request就可以发送合并请求了。

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: fzu2015/CourseManagement base: dev head fork: schaeper/CourseManage... compare: dev

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

Update README.md

Write Preview Styling with Markdown is supported

Leave a comment

Attach files by dragging & dropping, selecting them, or pasting from the clipboard.

Create pull request

Labels: None yet

Milestone: No milestone

Assignee: No one—assign yourself

当然，在发送请求之前，你可以检查一下你都改了哪些东西。在上面那个页面往下拉，就可以看到两者的对比。如下图

3 commits

1 file changed

0 commit comments

1 contributor

Commits on Nov 03, 2015

shaepher

Update README.md

081295a

shaepher

Update README.md

0598fbb

shaepher

Update README.md

3fdfe11

Showing 1 changed file with 2 additions and 2 deletions.

Unified

Split

4

README.md

这里显示你删除了哪些东西

这里显示你增加了哪些东西

<>

View

用红色表示

用绿色表示

1	# CourseManagement	1	# CourseManagement
2		2	
3	- fuz2015/docs这个repository已经删除，里面的文件重新发到该处的docs文件夹	3	+正在开发的教师报课系统
4		4	
5	-项目处于开发阶段，因此代码主要在dev分支。master分支等alpha版本发布的时候才会更新到dev分支的版本。	5	+由于本项目处于开发阶段，因此代码主要在dev分支。master分支等alpha版本发布的时候才会更新到dev分支的版本。

以上操作结束后，团队成员的流程就结束了。最后一步交给团队项目负责人来完成。

六、团队项目负责人审核及同意合并请求

首先进入GitHub的团队项目仓库中。此时右边的Pull requests显示当前项目有几个Pull request。点击进入查看。

fzu2015 / CourseManagement

Unwatch 1

Star 0

Fork 6

报课系统——团队 — Edit

25 commits

2 branches

0 releases

1 contributor

Branch: master CourseManagement / +

shaepher 添加dev分支说明

Latest commit c669fb5 2 days ago

CourseManagement

Update .gitignore

5 days ago

docs

Merge pull request #23 from fzu2015/docs

4 days ago

README.md

添加dev分支说明

2 days ago

<> Code

Issues 5

Pull requests 1

Wiki

Pulse

Graphs

选择一个Pull request

fzu2015 / CourseManagement

Unwatch 1

Star 0

Fork 6

Issues Pull requests Labels Milestones

Filters is:pr is:open

New pull request

1 Open 22 Closed

Author Labels Milestones Assignee Sort

Update README.md

#44 opened 39 seconds ago by shaepher

项目负责人审核有两个要注意的地方

- 一个是下图的①。一定要看清楚是合并到哪个分支。这里是从schaepfer的dev分支合并到fzu2015的dev分支。
- 另一个是下图的②。点击进去后，就可以查看该Pull request对项目做了哪些修改。这样如果有问题，可以及时发现，并关闭该Pull request。

如果关闭了，一定要告诉队友，否则他可能会不知道。虽然也可以直接在下面发布Comment告诉他，但队友不一定看到。

Update README.md #44

Edit

Open schaepfer wants to merge 3 commits into fzu2015:dev from schaepfer:dev ①

Conversation 0 Commits 3 Files changed 1 ② +2 -2

schaepfer commented 22 minutes ago Owner

No description provided.

schaepfer added some commits 26 minutes ago

- Update README.md 081295a
- Update README.md 0598fbb
- Update README.md 3fdfe11

Add more commits by pushing to the dev branch on schaepfer/CourseManagement.

✓ This branch is up-to-date with the base branch
Merging can be performed automatically.

Merge pull request ③ You can also [open this in GitHub Desktop](#) or view [command line instructions](#).

Labels None yet

Milestone No milestone

Assignee No one—assign yourself

Notifications **Unsubscribe**

You're receiving notifications because you authored the thread.

1 participant

- 如果没有问题，可以点击Merge pull request。这样就合并好了。

分支的问题

组员的本地仓库应该有两个分支，一个是main分支和dev分支。

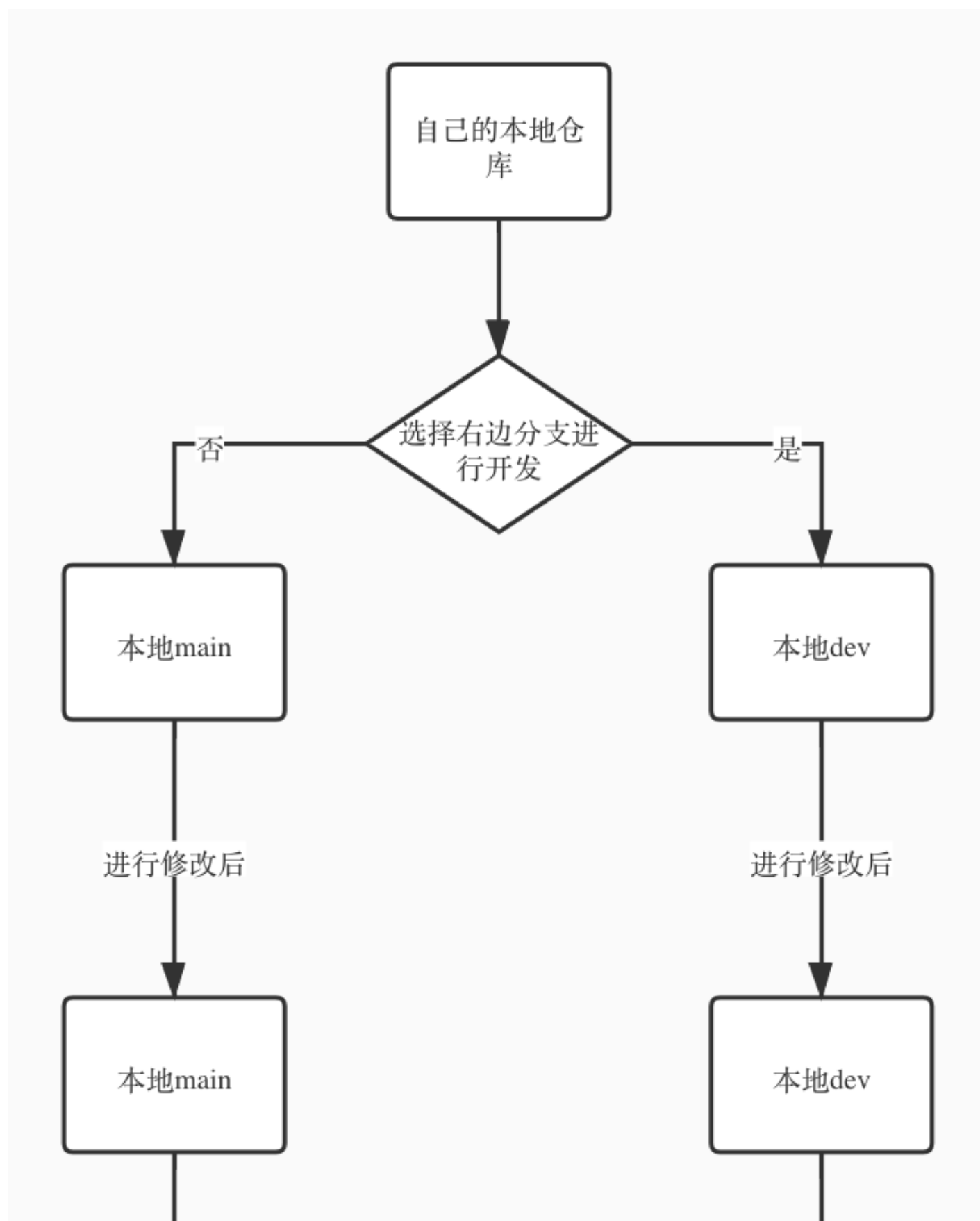
如果只在本地的dev分支上开发，开发完后，push到自己的远程仓库的dev分支上，然后pull到团队仓库上dev分支上。

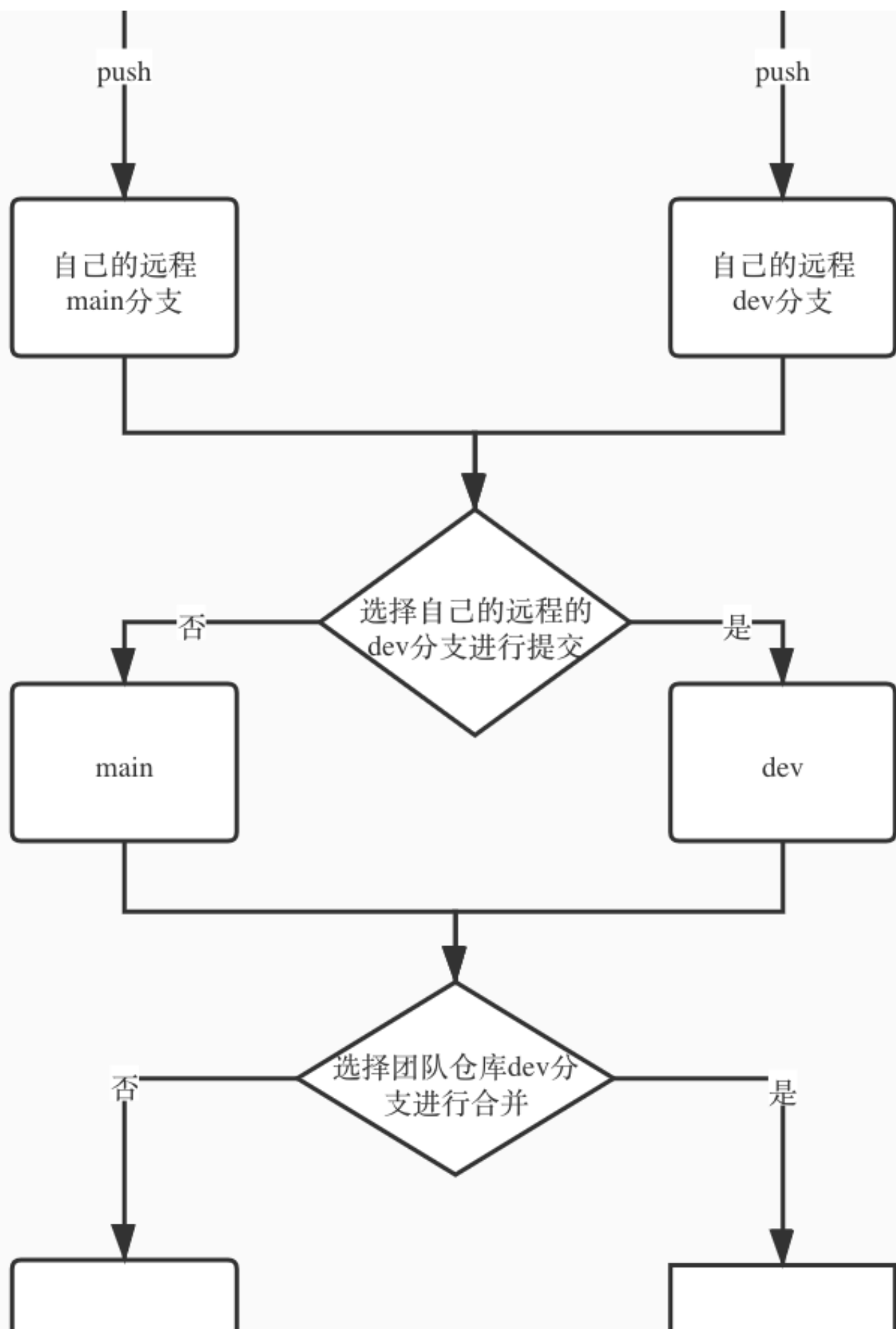
如果你在本地的dev分支上开发后，也可以先合并到本地的main分支上，此时你本地的main分支就有本地的dev分支的全部内容。

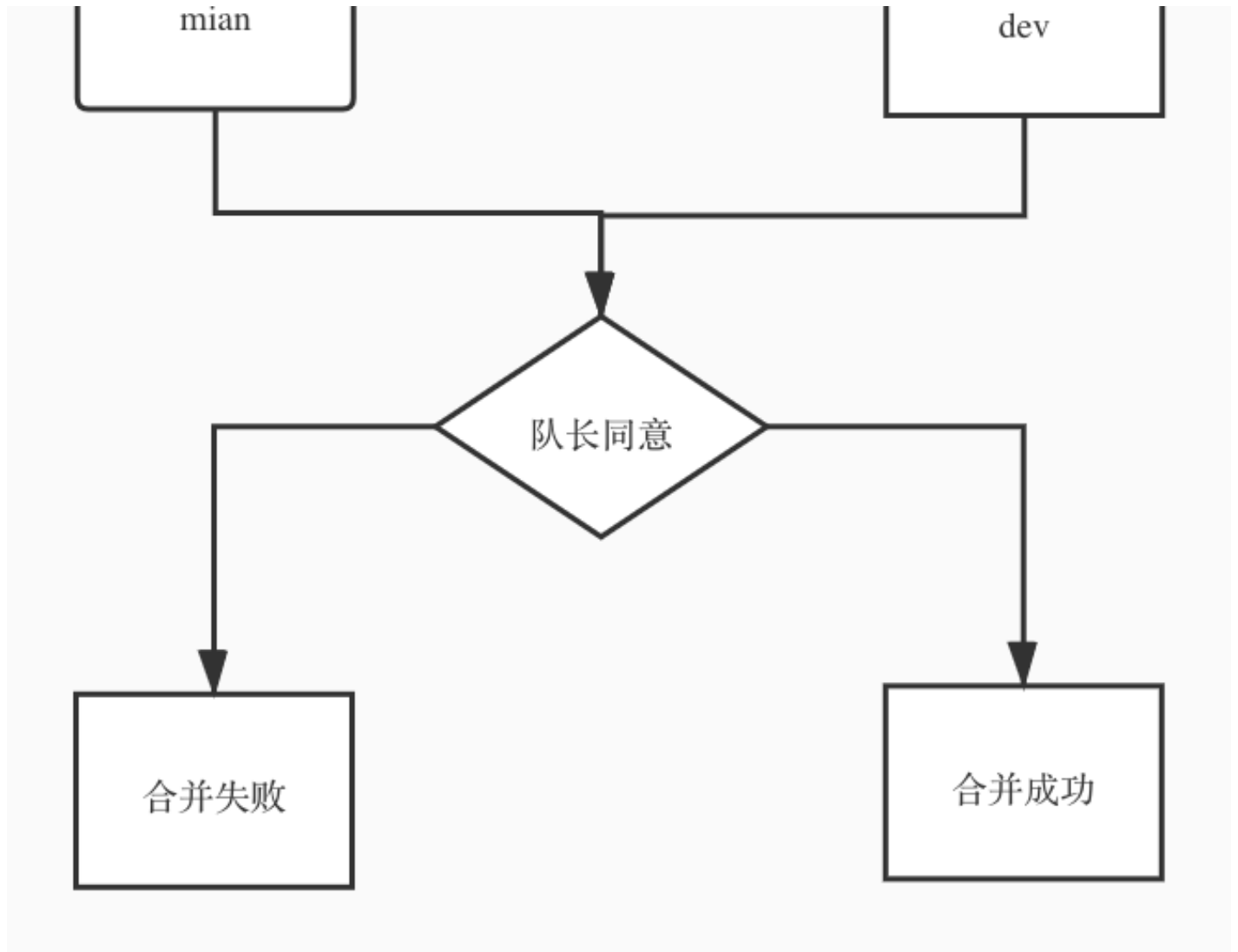
然后push到自己的远程仓库，你在哪个分支push的，就会推送到自己的远程仓库相对应的分支上。

最后，组员可以选择将自己的远程仓库的main分支还是dev分支pull到团队仓库的main分支还是dev分支。

组员pull后，队长那边是可以看到组员是以什么方式pull的，队长具有是否合并的权利。







如果你想要合并分支，无论是本地的，还是自己的远程仓库，还是团队的仓库，有以下的命令：

合并某分支到当前分支：`git merge <name>`

```
//比如你想把dev分支合并到main分支
git checkout mian//先切换到main分支
git merge dev//dev分支合并到main分支
//合并后，dev分支不会消失，main分支的内容会变的和dev分支一样；
```

重点强调!!!

自己的本地的仓库，自己的远程仓库，团队的仓库，都可以创建任意多的分支。

自己的本地的仓库，自己的远程仓库，团队的仓库，这三种仓库，都有分支的创建和合并的功能，如何使用，取决于你自己。

正常流程：

自己的本地仓库有main分支和dev分支，自己远程仓库有main分支和dev分支，团队的仓库也有main分支和dev分支。分支不过是不同的时间线，相当于两个平行宇宙，互不干扰。如果想要干扰，只能合并。

合并的方式也有很多种：

自己本地仓库main分支与dev分支合并。

自己远程仓库相当于自己的本地仓库的备份，自己本地仓库是什么样子的，自己的远程仓库就应该是什么样子的。

自己的远程仓库可以用main分支合并团队仓库的main或dev分支。

自己的远程仓库也可以用dev分支合并团队仓库的main或dev分支。

远程仓库能不能合并到团队仓库，取决于队长。

队长和组员的关系

队长和组员的操作大部分是一样的。

不过队长多了是否合并权，和创建团队组织和创建团队仓库。其他操作都和组员一样。

一篇文章

团队协作，为了规范，一般都是 fork 组织的仓库到自己帐号下，再提交 pr，组织的仓库一直保持更新，下面介绍如何保持自己 fork 之后的仓库与上游仓库同步。

下面以我 fork 团队的博客仓库为例

点击 fork 组织仓库到自己帐号下，然后就可以在自己的帐号下 clone 相应的仓库

使用 `git remote -v` 查看当前的远程仓库地址，输出如下：

```
source-shell
origin  git@github.com:ibrother/staticblog.github.io.git (fetch)
origin  git@github.com:ibrother/staticblog.github.io.git (push)
```

可以看到从自己帐号 clone 下来的仓库，远程仓库地址是与自己的远程仓库绑定的（这不是废话吗）

接下来运行 `git remote add upstream https://github.com/staticblog/staticblog.github.io.git`

这条命令就算添加一个别名为 upstream（上游）的地址，指向之前 fork 的原仓库地址。`git remote -v` 输出如下：

```
origin  https://github.com/snowfigure/solo.git (fetch)
origin  https://github.com/snowfigure/solo.git (push)
upstream      https://github.com/b3log/solo.git (fetch)
upstream      https://github.com/b3log/solo.git (push)
```

之后运行 `git pull upstream` 就可以保持本地仓库和上游仓库同步了

之后运行下面几条命令，就可以保持本地仓库和上游仓库同步了

```
git fetch upstream  
git checkout master  
git merge upstream/master
```

接着就是熟悉的推送本地仓库到远程仓库

```
git push origin master
```