1.创建团队的项目

- 参考文档：https://www.cnblogs.com/thousfeet/p/7840932.html
- 参照官方文档从头创建一个新的组织 LuFeiTeam
  - settings-Organizations-New organizations
- 进入组织。邀请TestsDC(提前创建好的另一个GitHub用户)加入组织。
- 进入组织。创建新团队 Teams-New team（student）。并邀请组织成员加入进刚创建好的新团队student。
- 进入组织。创建新项目(LuFeiTeam组织的项目)
  - Public任何人都看得到,但不是任何人都有写的权限;Private保管私有的仓库
  - 勾选初始化生成readme.md和.gitignore文件;以及许可证的类型(MIT License)
- 进入刚创建好的项目。
  - 设置-管理访问-添加刚创建的LuFeiTeam/student团队到此项目中并赋予read权限.
  - setting-Manage access-invite teams or people

2.如何选择适合自己团队的工作流?

团队对研发项目的分工,最终要形成一个公共的产品,势必会涉及到协作与集成的环境。即对于研发团队而言的一个分支管理的流程。

- 主干开发: eg 谷歌、facebook. （国内不常见）
  - 任何人一天当中的变更产生的commit会很及时的融入到公司的公共仓库里面,其它成员能够第一时间看到每个成员的变更。
  - 适用于: 开发团队系统设计和开发能力强。有一套有效的特性切换的实施机制(哪些功能在线上起作用)。
  - 适用于: 组件开发的团队,成员能力强,人员少,沟通顺畅。
- Git Flow: 很繁琐
  - 属于特性分支开发的方式
  - 适用于: 不具备主干开发能力。有预定的发布周期。需要执行严格的发布流程!
- GitHub Flow
  - 适用于: 不具备主干开发的能力。随时集成随时发布: 分支集成时经过代码评审和自动化测试,就可以立即发布的应用。
  - master作为主线(现在是main).有一个功能开发的话,基于main上的某一个里程碑点拉一个特性分支出去,开发到一定程度后再合成到master上。
  - 合成来后,基本上就可以上线了,不用等待。适合只有一个版本的服务。
- GitLab Flow（带生产分支 master->production）
  - 基于上面github的流程做了一个完善。
  - 适用于: 不具备主干开发能力。无法控制准确的发布时间,但又要求不停的集成。
  - 从master上拉的特性分支开发完后集成到master上后,不是马上就上线的。这时候需要拉一个PRODUCTION分支(用作上线)做一个缓存,为了不影响master的持续集成。
- GitLab Flow（带环境分支 master->pre-production->production）
  - 适用于: 不具备主干开发能力。需要逐个通过各个测试环境的验证才能发布。
- GitLab Flow（带发布分支）
  - 适用于： 不具备主干开发能力。需要对外发布和维护不同版本。
  - 像在传统行业里,视频会议等设备要跟着硬件走的,硬件不会实时升级,所以软件这块会有2.3、2.4版本等。
  - 同一个时间点会有多个发布分支。（2-4-STABLE -> MASTER -> 2-3-STABLE）

3.如何挑选合适的分支集成策略?

涉及到具体几个分支做集成的时候,存在哪些集成的策略。这些集成策略在github上对每个仓库都有相应的设置配置项。

- Fork GitGarden/git_with_travelling 到自己组织项目中,并加入组织已创建好的团队
- 进入此项目 - Insights - Network 可以看到整个版本树的演进。
  - 为了搭建分支的集成,基于同一个点创建了Beijing和Shanghai两个特性分支,最终要作为整体合并到master上面去。
  - 两个特性分支,肯定有人先提交,有人后提交,关于分支的集成策略.
- 进入此项目 - settings - Merge button 合并按钮策略
  - Allow rebase merging
    - 严格的线性方式,希望最后的历史树就是一条很清晰的主干线
    - 从主干上拉出特性分支开发,回去的时候必须基于主干的head,把特性分支里面的commit基于主干最新的commit做rebase
    - rebase结束后,完全可以将特性分支删除掉,留下纯粹的逻辑很清晰的一条主线
  - Allow squash merging
  - Allow merge commits
- 注意:公共的集成分支是不允许被强制的执行push -f的