转自：[在团队中使用GitLab中的Merge Request工作模式](https://link.zhihu.com/?target=http%3A//www.fwhyy.com/2018/06/Use-the-Merge-Request-working-mode-in-GitLab-in-the-team/)

在工作中使用Git已有5年多的时间了，Git分布式的工作机制以及强大的分支功能使得在团队中推广使用没有受到什么阻碍。一直以来都是采用的分支管理模式，可以把项目的开发分为四个阶段：开发、测试、预发布和发布。

## 分支管理模式

## 开发阶段

1. 除了master分支创建一个供所有开发人员开发的dev分支；
2. 开发人员在dev分支上进行工作，随时随地commit，每天push一次到服务器；
3. push代码前需要进行pull操作，因为有可能在之前有别的成员先进行了push操作，如果有冲突还需要进行冲突解决；
4. 每天上班后所有成员对dev进行pull操作，获取所有成员push的代码，有冲突需要解决；
5. 团队Leader每天将dev合并一次到master。

## 测试阶段

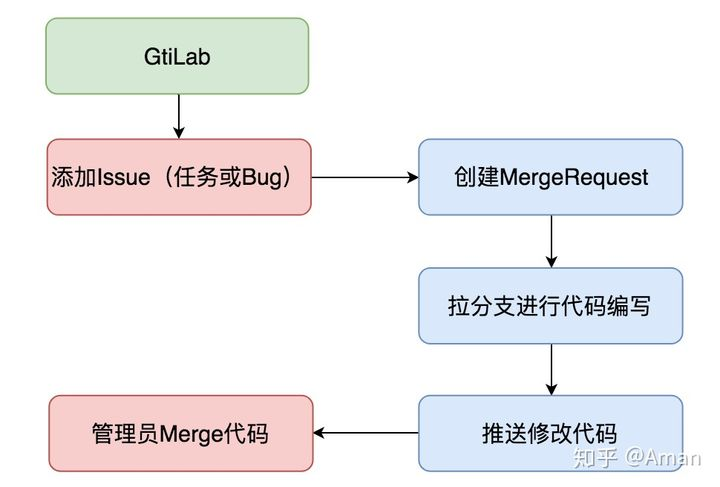
1. 测试进入后就需要添加test分支；
2. 在开发人员将代码push到dev分支后，可以在dev基础上创建test分支，测试人员以test分支搭建测试环境，开始测试；
3. 开发人员在接受到bug后，直接在测试分支上修改，然后让测试人员进行验证；
4. 每天团队Leader将测试分支上修改的bug合并到dev分支上，这样所有团队成员当天修复的bug都会在第二天被团队其他人pull下来；
5. 团队Leader每天将dev合并一次到master。

## 预发布和发布阶段

1. 系统上线后试运行阶段会存在两种改动：bug和优化需求，bug通常当天解决晚上部署，优化需求通常周末部署；
2. bug当天能修复的就直接在test分支上修复，然后进行测试，验证通过后合并到master；
3. bug当天不能修复的就针对该bug创建一个分支，修复完后合并到test分支进行测试，验证通过后合并到master；
4. 每个优化需求都以master分支为基础创建一个feature分支，完成后合并到dev分支，开发人员可以先交叉测试，然后将dev合并到test进行测试，验证通过后合并到master；
5. master始终是一个干净的，可发布的分支。

## Merge Request模式

一直以来，都觉得Merge Request模式遥不可及，只有做开源软件才会采用这种模式，没想到这么快就已经在团队中开始推行使用了，先看一张图来了解下Merge Request的开发流程：

需求或是Bug都是用Issue来表示；

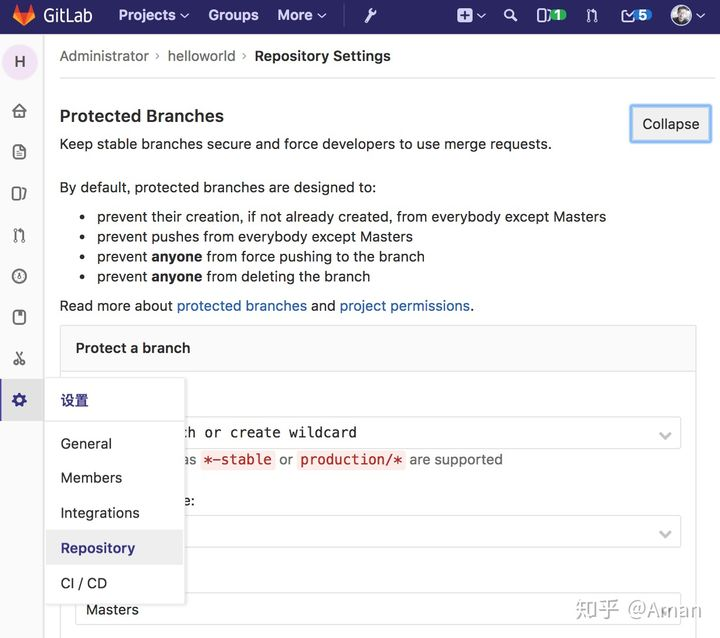
1. 虽然Issue不支持多层级，但结合里程碑、标签等还是可以很好的对任务和Bug进行管理；
2. 管理员和团队成员都可以进行Issue的创建；
3. 任务的接收者对Issue创建Merge Request；
4. 完成任务后推送代码到Merge Request对应的分支；
5. 管理员对代码进行Merge。

**相比较传统的分支管理模式，Merge Request可以给我们带来下面几个好处：**

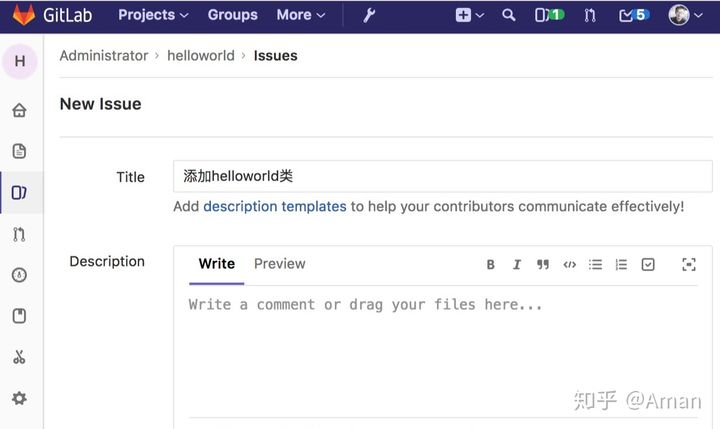
1. 重要分支设置为受保护，杜绝了有些问题代码被提交了，但项目经理不知道的情况；
2. 每个任务都有一个对应的分支，互相隔离，所有的代码改动有据可查；
3. 开发人员不用在分支间切换和合并，更专注于开发。

下面以一个示例来介绍Merge Request的工作流程

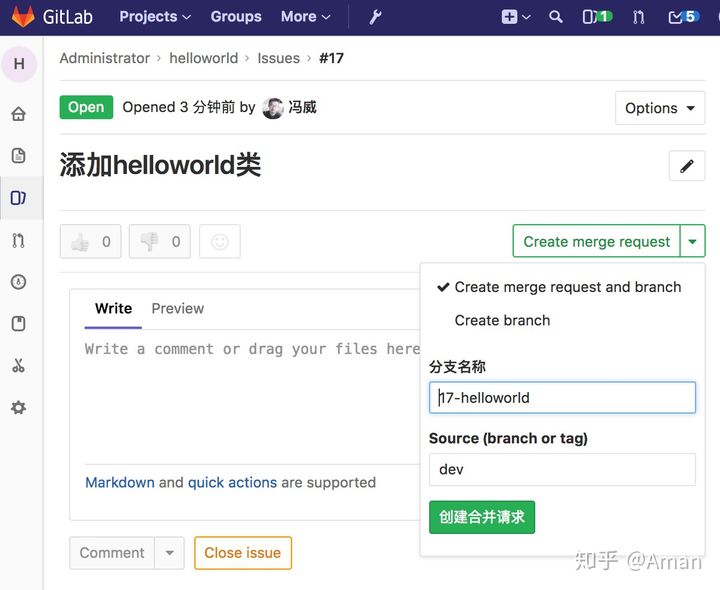
1、设置重要分支受保护

在上图中的位置可以将所有的重要分支设置为受保护，重要的分支通常是master、release、test等。

2、创建Issue

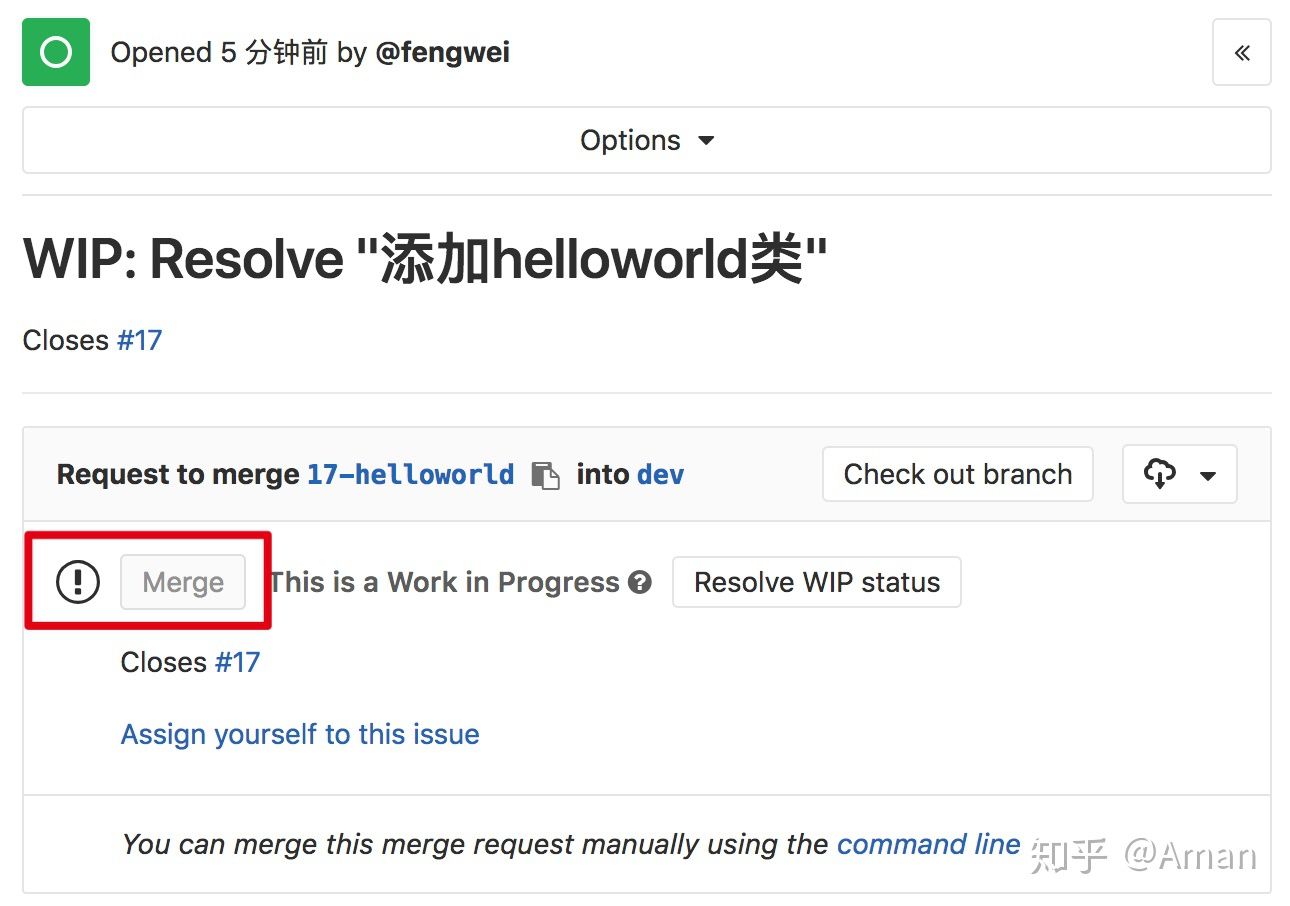


任务创建后，开发人员就可以对该任务创建Merge Request了，如下图：

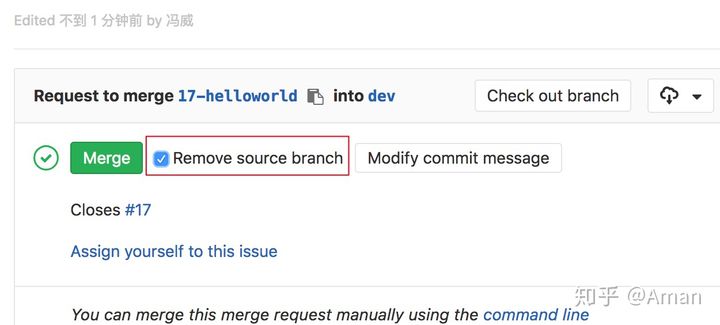


* 创建Merge Request时会创建针对这个任务对一个分支；
* 分支名称的格式为：任务编号-[任务标题中出现的英文和数字]，当然分支名称也可以自行修改；
* 分支的Source为该项目设置的主分支，主分支可以在设置/General/General project settings/Default Branch进行设置。

3、使用你熟悉的工具拉取Merge Request对应的分支到本地进行代码修改，修改完成后，Push代码到服务器，代码推送后，管理员在Merge Request页面可以看到Merge按钮，如下



点击右边的Resole WIP status后，Merge按钮就可以使用



如果勾选Remove source brance，当Merge后，服务器端会删除创建的分支。Merge完成，会关闭关联的任务，但并不是每一次推送都可以非常顺利，有时会有冲突，当本地代码和服务器代码不一致时，会出现解决冲突的按钮，解决冲突后才能进行Merge



代码Merge后，开发人员就可以按照同样的流程做下一个任务了。

## 思考

* 如果系统上线后有紧急Bug需要处理，这个流程应该怎样去调整？
* 每个任务都在单独分支并行开发，这是如果A和B都以来C开发的一个模块，应该怎么解决？
* 理论上Issue管理员和开发人员都可以进行创建，什么样的Issue可以有开发人员来创建？