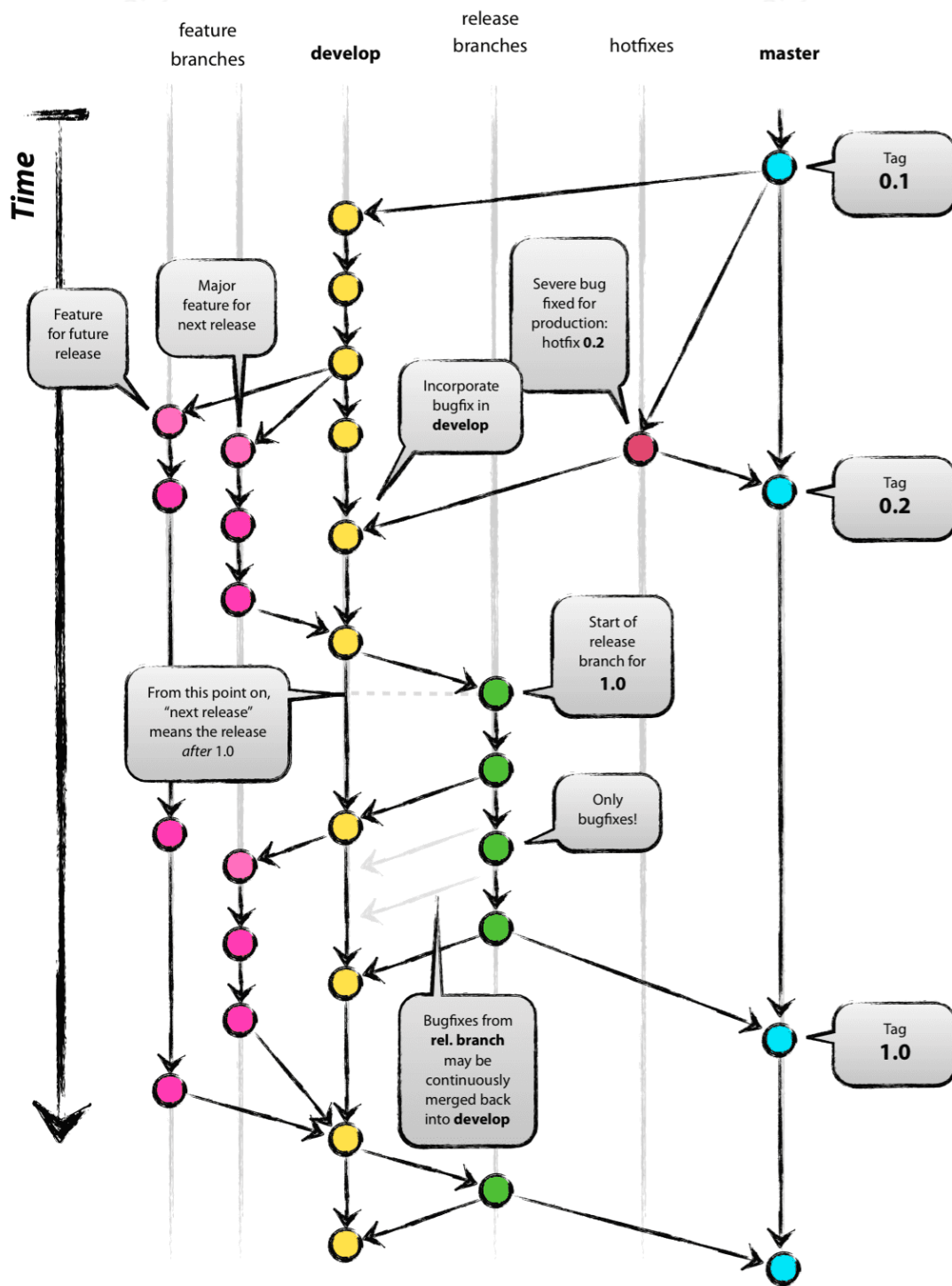


基于Git flow定制专属workflow

git flow简介

Git flow是一种以发布为中心的开发模式，其工作流如下：



该流程以分支名表示软件开发中开发状态的迁移。

1. 从开发分支（develop）创建特性分支（feature branches），进行功能的实现或修改
2. 特性分支（feature branches）的修改结束后，与开发分支（develop）合并
3. 重复上述步骤2，不断实现功能直至可以发布
4. 创建用于发布的分支（release branches），处理发布的各项工作

5. 发布准备工作完成后与 master 分支合并，打上版本标签（Tag）进行发布
6. 如果发布的软件出现 BUG，以打了标签的版本为基础进行修正（hotfixs）

上述的流程十分的完备，但是对于我们目前的实际项目来说，有很多其实是多余的，所以我们需要根据git flow定制专属于我们自己的workflow。

项目现状

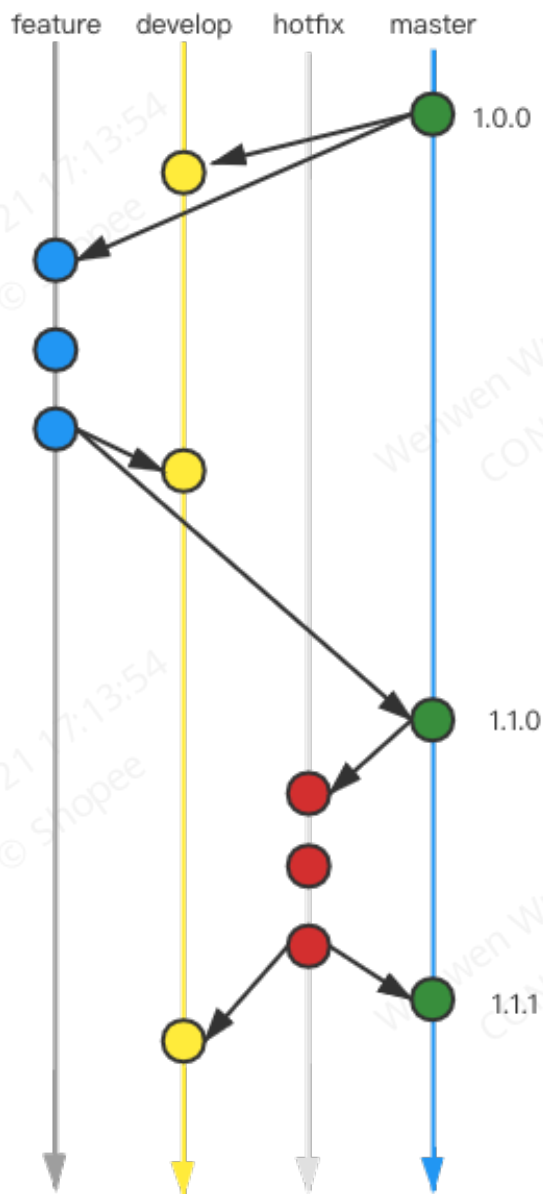
我们目前的开发现状是有多个feature在同时开发，同时还会有不定时的hotfix需要发布，但是我们的功能很多都需要发布到测试环境才能测试，所以我们需要一个集成分支，集成开发中的feature，方便进行测试。我们目前没有专职的测试人员，所以测试依赖于开发，所以我们没有release相关的分支，master就是上图中的release分支的作用。

定制workflow

首先说明诉求：

1. 需要集成分支，方便测试
2. 发布要简单，有一个始终是可以发布状态的分支：master/main分支

定制的workflow如下：



开发流程说明：

1. 新版本开发时，从master新建feature分支，命名规范是release-{feature-name}-{version}
2. 在feature分支上面开发
3. 在需要部署测试环境是，将feature分支的更改merge到test分支
4. 重复步骤2、3，直到测试通过等待上线
5. 提交feature分支合并到master的Pull Request (PR)
6. 合并PR后在master分支打tag，然后发布，然后删除release分支

在这个workflow中，feature和hotfix开发的流程是一样的，都是从master新建分支，在test分支验证功能，验证完成后提交PR将改动点合并到master，最后使用master的最新代码部署到生产环境。

说明

master：最新的线上可运行版本，可用于rollback

hotfix：从master分支新建最新的hotfix分支，验证完成后merge到master，merge后删除

feature：从master分支新建最新的开发分支，merge到master后删除

环境：test环境、生产环境

参考文章

1. A successful Git branching model: <https://nvie.com/posts/a-successful-git-branching-model/>
2. Understanding the GitHub flow: <https://guides.github.com/introduction/flow/>
3. git 工作流程: <https://www.ruanyifeng.com/blog/2015/12/git-workflow.html>