Git做分支你们是怎么管理的

相关文章
Git日常开发常用命令汇总

文章目录

前言

记得刚工作的时候根本不知道什么是版本管理工具，有一次和别人聊天，人家问你们公司代码用什么版本管理工具？我说啥是版本管理工具，我们一般用U盘拷贝，然后人家就顾左右而言他了。后来我知道了有个东西叫SVN,后来又知道了还有个东西叫Git。所以说刚毕业的同学一定要优先进入专业的大公司，就像年轻时候应该去大城市闯两年一样，眼界以及你遇到的牛人会大大加快你以后成功的进程。

概述

本文主要是介绍一种在具体实践中使用Git来管理项目开发的一种成功的方式，其实主要思想来源于这篇文章 A successful Git branching model，网上大部分教程都是致敬这篇文章。

Git的基本使用方法

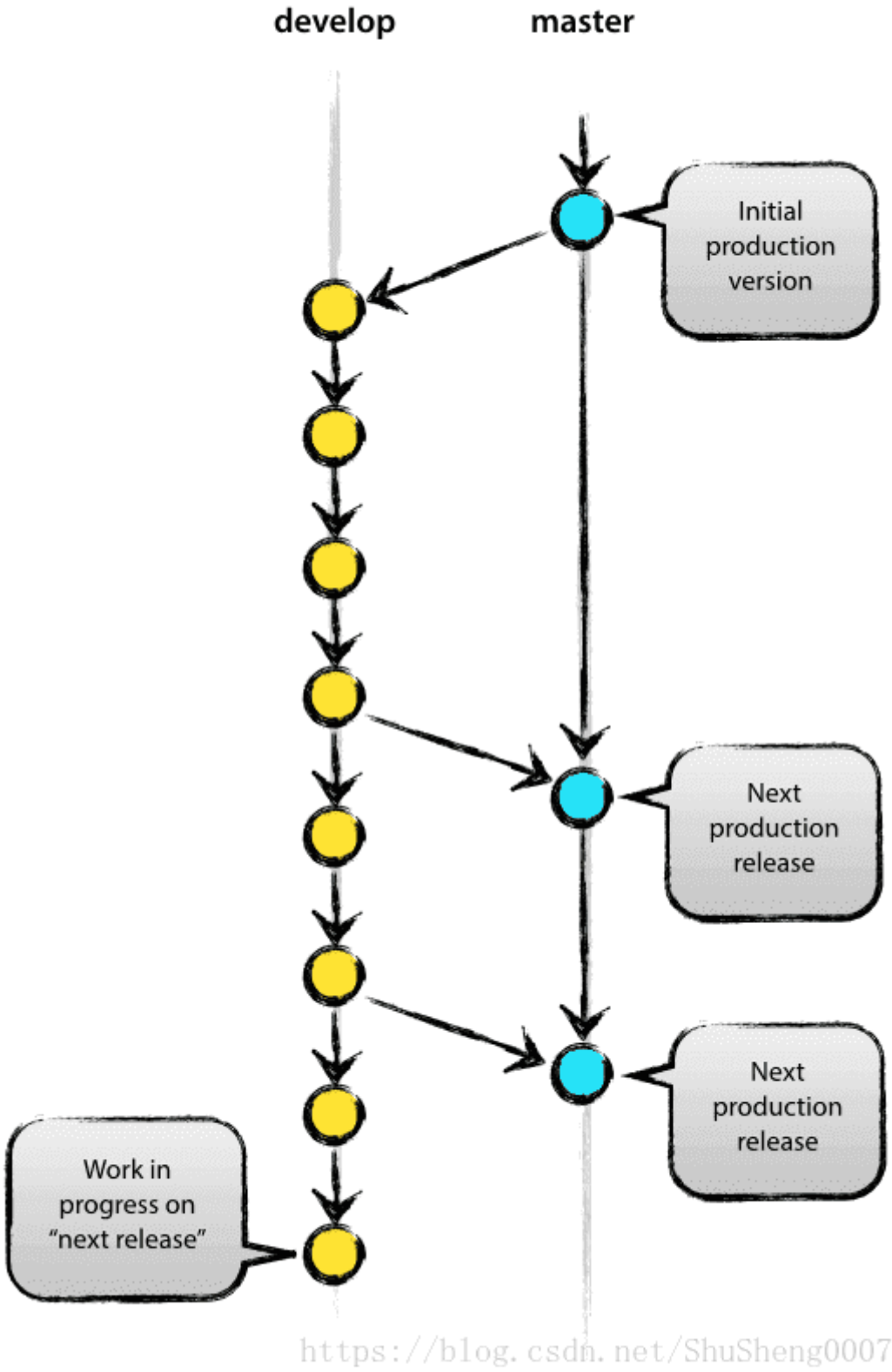
关于git的基本教程，强烈建议阅读廖雪峰老师的Git教程，对初学者非常友好。

使用Git管理项目的方式

在实际开发中如何使用Git没有一个标准答案，使用方式也是各式各样，很多基本上都是把Git当SVN来用。下面介绍的是一种经过实践的运行比较良好的管理方式。

主分支

实际开发中，一个仓库（通常只放一个项目）主要存在两条主分支：master与develop分支。这个两个分支的生命周期是整个项目周期。就是说，自创建出来就不会删除，会随着项目的不断开发不断的往里面添加代码。master分支是创建git仓库时自动生成的，随即我们就会从master分支创建develop分支，如下图所示。



master：这个分支最为稳定，这个分支代表项目处于可发布的状态。

> 例如王二狗向master分支合并了代码，那就意味着王二狗完成了此项目的一个待发布的版本，项目经理可以认为，此项目已经准备好发布新版本了。所以master分支不是随随便便就可以签入代码的地方，只有计划发布的版本功能在develop分支上全部完成，而且测试没有问题了才会合并到master上。

develop：作为开发的分支，平行于master分支。

> 例如王二狗要开发一个注册功能，那么他就会从develop分支上创建一个feature分支 fb-register（后面讲），在fb-register分支上将注册功能完成后，将代码合并到develop分支上。这个fb-register就完成了它的使命，可以删除了。项目经理看王二狗效率很高啊，于是：“二狗你顺带把登录功能也做了吧”。二狗心中暗暗骂道：日了个狗的，但是任务还的正常做，二狗就会重复上面的步骤：从develop分支上新创建一个名为fb-login的分支，喝杯咖啡继续开发，1个小时后登录功能写好了，二狗又会将这个分支的代码合并回develop分支后将其删除。

通过以上分析可以发现，我们可以使用Git hook 脚本自动发布发布新的版本，具体就是每当有代码从develop分支合并到master分支的时候，脚本就会自动触发，编译发布新的版本。

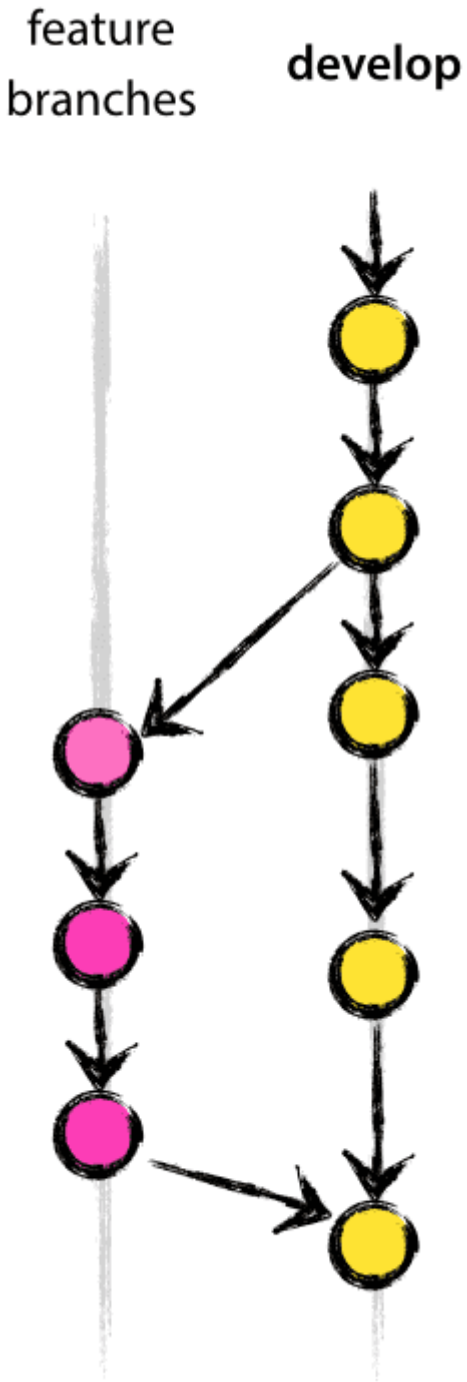
支持分支

这些分支都是为了程序员协同开发，以及应对项目的各种需求而存在的。这些分支都是为了解决某一个具体的问题而设立，当这个问题解决后，代码会合并回主分支develop或者master后删除，一般我们会人为分出三种分支。

Feature branches：

> 这种分支和我们程序员日常开发最为密切，称作功能分支。 必须从develop分支创建，完成后合并回develop分支

如下图所示。



具体事例可以参考上面王二狗完成登录注册功能时的做法。

Release branches:

> 这个分支用来分布新版本。

从develop分支创建，完成后合并回develop与master分支。

这个分支上可以做一些非常小的bug修复，当然，你也可以禁止在这个分支做任何bug的修复工作，而只做版本发布的相关操作，例如设置版本号等操作，那样的话那些发现的小bug就必须放到下一个版本修复了。如果在这个分支上发现了大bug，那么也绝对不能在这个分支上改，需要Featrue分支上改，走正常的流程。

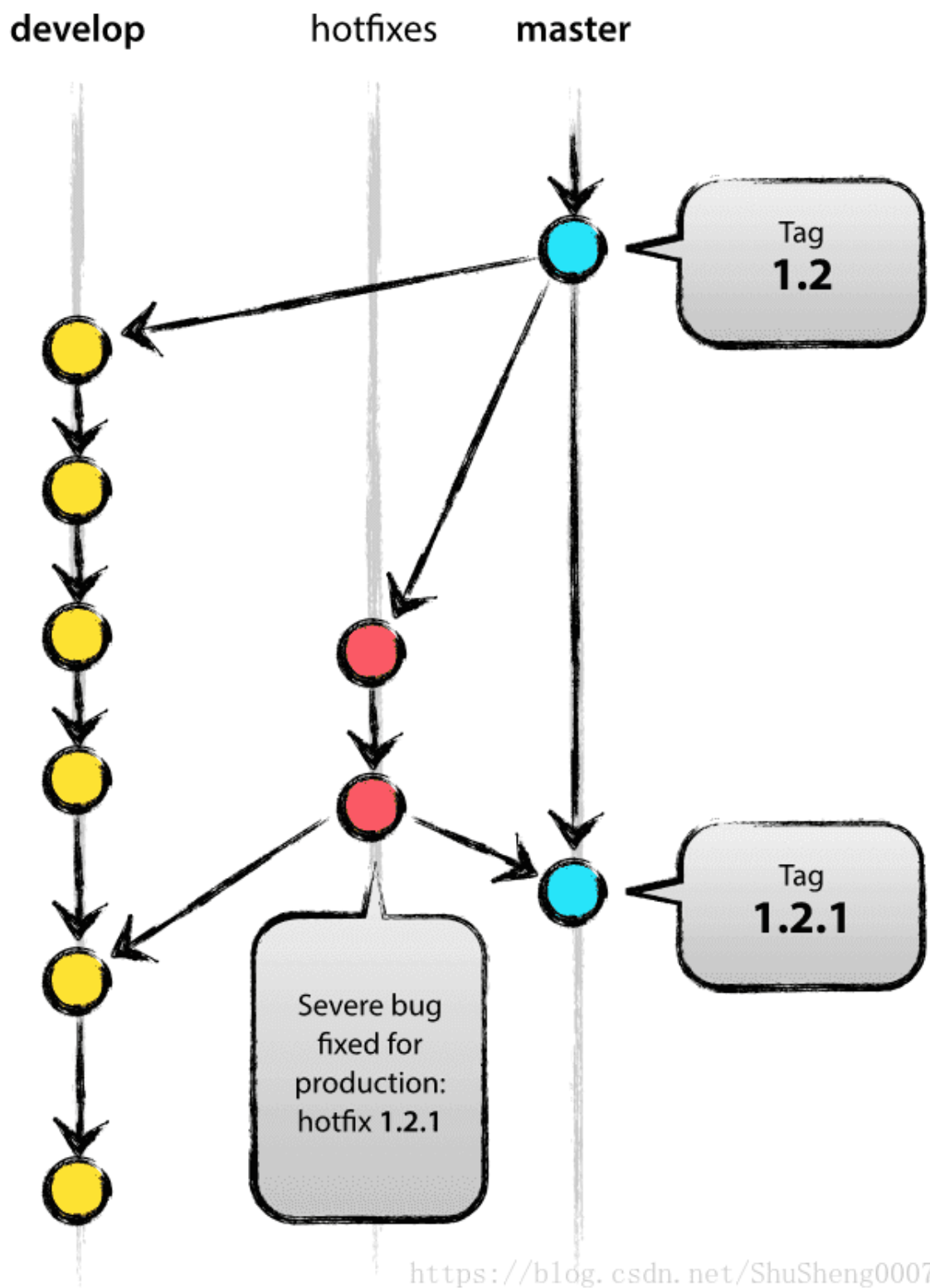
实例：王二狗开发完了登录注册功能后决定发一个版本V0.1，那么他先从develop分支上创建一个Release 分支release-v0.1，然后二狗在这个分支上把版本号等做了修改。正准备编译发布了，项目经理说：“二狗啊，你那个登录框好像向右偏移量1个像素，你可以调一下吗？”二狗心中有暗暗骂道：日了个狗，但是。。。你们懂得，功能还的正常改。不过二狗发现这个bug特别小，对项目其他部分不造成

不可预知的问题，所以直接在release分支上改了，然后愉快的发布了版本1.0.版本上线后，二狗将这个分支分别合并回了develop与master分支，然后删除了这个分支。

Hotfix branches：

> 这个分支主要为修复线上特别紧急的bug准备的。
>
> 必须从master分支创建，完成后合并回develop与master分支。

如下图所示：



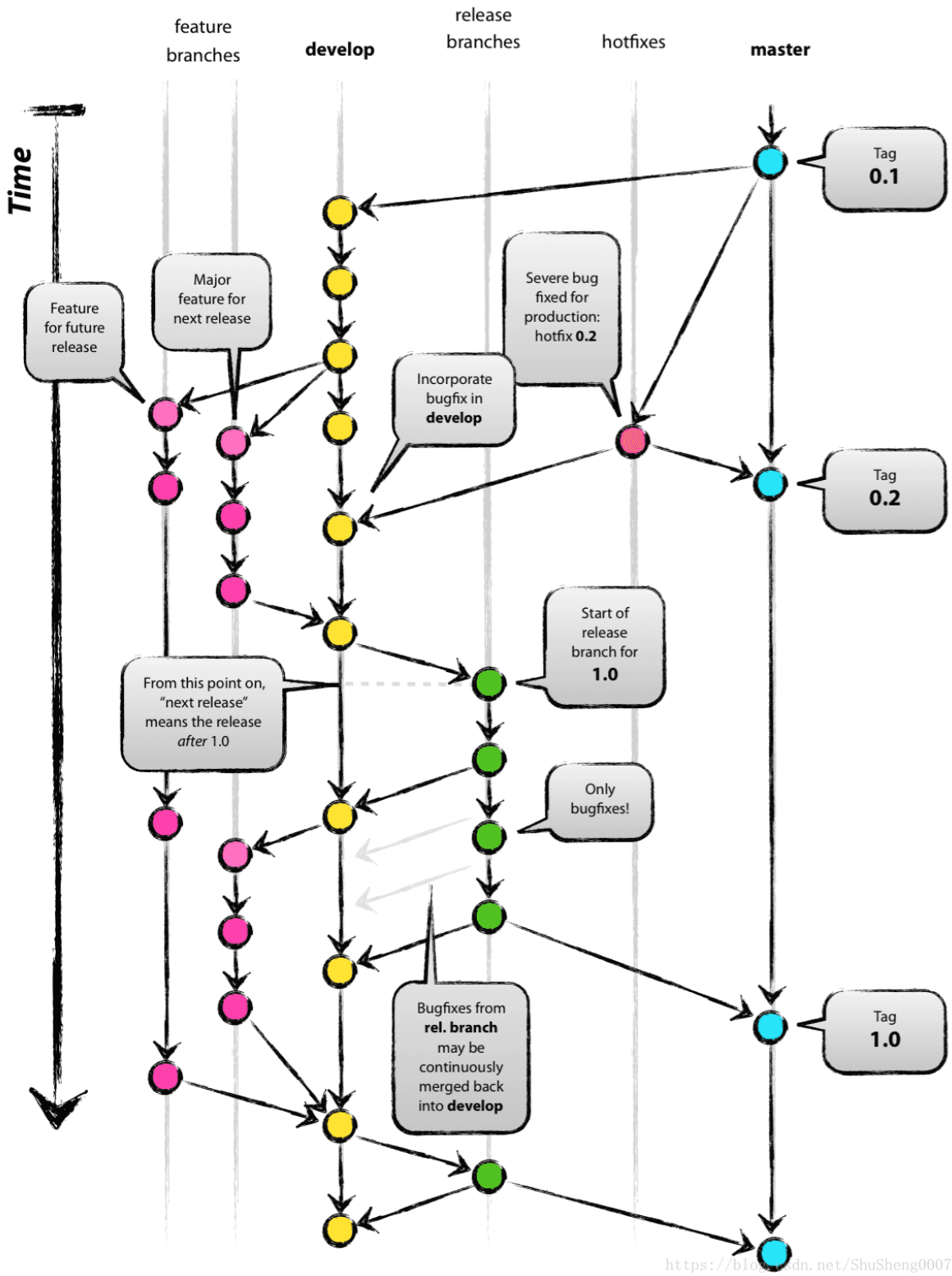
这里写图片描述
这个分支主要是解决线上版本的紧急bug修复的，例如突然版本V0.1上有一个致命bug，必须修复。那么我们就可以从master 分支上发布这个版本那个时间点 例如 tag v0.1（一般代码发布后会及时在master上打tag），来创建一个 hotfix-v0.1.1的分支，然后在这个分支上改bug，然后发布新的版本。最后将代码合并回develop与master分支。

实例：

某天夜里二狗正在和女朋友嘿咻呢，突然项目经理打来电话：“二狗啊，线上出了个大问题，大量用户无法登录，客服电话已经被打爆了，你紧急处理一下”。二狗心中默默骂道：日了个狗，然后就爬起来去改代码了。二狗先找到master分支上tag v0.1 的地方，然后新建了hotfix-v0.1.1 分支，默默的修bug去了。经过一个多小时的奋战，终于修复了，然后二狗改了版本号，发布了新版本。二狗将这个分支分别合并回了develop与master分支后删除了这个分支。终于搞定了，回头看看床上的女票已经进入了梦乡，二狗贼心不死，上前挑逗，此刻女票将满腹怨气集于一点，使出凌空一脚将其登于床下，随后甩一枕于二狗，二狗会意，趋客厅，抱枕卧于沙发。。。

总结图

上面的讲解最后汇成一张图



总结

希望广大程序员不要有王二狗的悲惨遭遇，最后希望广大“王二狗媳妇”可以理解广大“王二狗”的苦衷。

最后，求关注，求点赞！有任何疑问可以评论留言，我会尽力回复的。