

GitFlow流程



茉莉儿



0.336



2018.08.24 18:56:18



字数 2,607



阅读 1,035

我所有的文章都会在我的[小站](#)同步更新，有问题可以留言给我

关于gitflow

概念：

Git Flow是构建在Git之上的一个组织、管理软件开发活动的模型。Git Flow是一套使用Git进行源代码管理时的一套行为规范和，通过利用Git创建和管理分支的能力，为每个分支设定具有特定的含义名称，并将软件生命周期中的各类活动归并到不同的分支上。实现了软件开发过程不同操作的相互隔离。这种软件开发的模型被称为“Git Flow”。

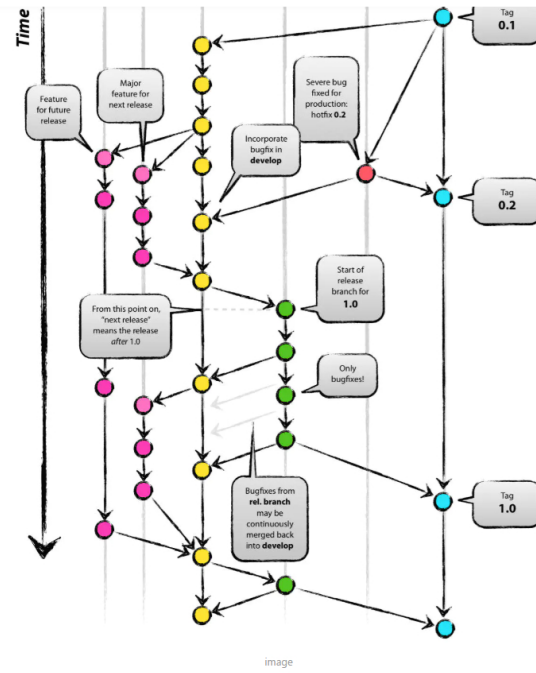
原理：

gitflow的核心就branch，通过在项目的不同阶段对branch的不同操作包括但不限于create、merge、rebase、等来实现一个完整的高效率的工作流程。一般而言，软件开发模型有常见的瀑布模型、迭代开发模型、以及最近出现的敏捷开发模型等不同的模型。每种模型有各自应用场景。Git Flow重点解决的是由于源代码在开发过程中的各种冲突导致开发活动混乱的问题。因此，Git Flow可以很好的于各种现有开发模型相结合使用，尤其是多人合作开发时提高效率。用一张图来了解gitflow的流程：从右向左看 从上到往下看

feature branches develop release branches hotfixes master

写下你的评论...

评论0 赞3 ...



Branch:

Branch是gitflow的核心。主要分为两大类 Main Branches 和 Supporting branches, 其中 Main Branches 中又包含了 Master 和 Develop, 而 Supporting branches 中包含了 Feature、Release、Hotfix 以及其他自定义分支。下面逐一讲解：

Master:

• 描述：

master分支上存放的是最稳定的正式版的代码，并且该分支的代码应该是随时可在开发环境中使用的代码（Production Ready state）。当一个版本开发完毕后，产生了一份新的稳定的可发布的代码时，master分支上的代码要被更新，同时，每一次更新，都需要在master上打上对应的版本号(tag)。

• 生成及销毁：

任何人不允许在master上进行代码的直接提交，只接受合入，Master上的代码必须是从经过多轮测试且已经发布一段时间(根据DAU以及项目实际情况来定，个人建议K歌国际版可以定为一周)且线上已经稳定的 release 分支合并进去，然后在Master 上生成tag(通常就是对应的版本号)

• 命名：

master

Develop



茉莉儿

总资产15 (约1.43元)

关注

招收iOS开发者啦!!!

阅读 245

SVN切换到Git方法及坑点

阅读 4,066

推荐阅读

iOS高级开发工程师-面试-笔试

阅读 2,358

docker轻量级管理工具

阅读 3,920

js:Promise的简单理解

阅读 14,023

前端面试之简历篇

阅读 14,665

备战CKA每日一题——第0天(考前准备)

阅读 62

develop.

• 描述:

develop分支是保存当前最新版本开发成果的分支。该分支上的代码允许有BUG，但是必须保证编译通过，且该分支可以作为每天夜间测试的分支(如果有夜间测试的话)所以该分支也叫做 Nightly build。当develop分支上的代码已实现了软件需求说明书中所有的功能(必须经过开发自测，但是不必经过QA)且相对稳定时候，就可以基于此分支来拉出新的release分支交付QA进行测试。

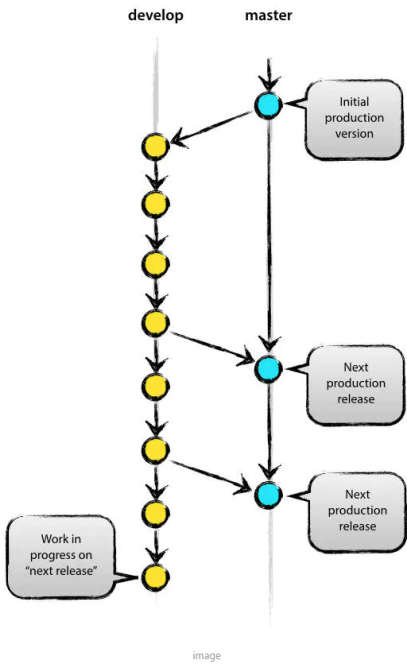
• 生成及销毁:

Develop分支是由一个人(通常是Team Leader)从Master中拉出，任何人不得在Develop上进行代码提交，只接受合并。Develop上所有代码一定都是由 Supporting branches 中的Branch合并进来，且合并Develop的分支必须保证功能完整，可以独立运行，可允许包含一些BUG(但是最好经过自测，不要有太大或者太明显的BUG，比如一启动就crash之类的)。

• 命名:

develop

• 流程:



Feature:

• 描述:

Feature分支通常叫做功能分支，也可以叫做个人分支，一般命名为 **feature/XXXX**。该分支就是每一个开发人员进行开发的分支，比如做一些功能、需求之类的东西，这个分支上的代码变更最终合并回develop分支或者干脆被抛弃掉（例如实验性且效果不好的代码变更）。一般而言，feature分支代码可以保存在开发者自己的代码库中而不强制提交到主代码库里。

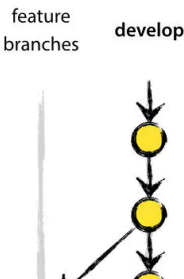
• 生成及销毁:

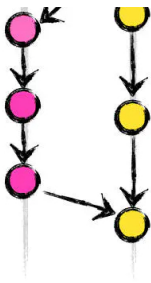
每个开发者从通常会Develop分支中拉取自己的feature，且开发者可以随意的在自己的feature上进行操作 包括但不限于 提交、回滚、删除。如果最终需要合并入develop那就要保证功能的完整性以及代码的稳定新，比如我在feature上做了3个需求但是由于时间关系我只做了两个，那也可以将feature合并入develop，然后剩下的那一个需求等有时间了再去feature上做完之后再合并develop。所以这里说的功能的完整性并不是值得要做完所有的功能，而是要保证你所要做的所有需求中的某一个或者某几个功能已经做完，不允许把做到一半的功合并入develop。合并入develop尽量上删除远端的feature分支，本地的feature可以视情况而取舍。

• 命名:

feature通常是从develop上拉取 所有通常用 **dev_功能描述_英文名** 来命名。比如 **feature/dev_refresh_molierzhang**

• 流程:





image

Release:

• 描述:

Release分支通常叫做发布分支,也可以叫做测试-发布分支,一般命名为 **Release/1.2.3** (后面是版本号),该分支是为测试-发布新的产品版本而开辟的。因为包含测试流程,所以在这个分支上的代码允许做小的缺陷修正、准备发布版本所需的各项说明信息(版本号、发布时间、编译时间等等)。通过在release分支上进行这些工作可以让develop分支空闲出来以接受新的feature分支上的代码提交,进入新的软件开发迭代周期。**注意:该分支上的代码一定是可编译可运行的,允许包含小BUG**

• 生成及销毁:

当develop分支上的代码已经包含了该版本所有即将发布的功能和需求,并且已通过自测且已基本稳定,我们就可以考虑准备基于develop拉取release分支了。而所有在当前即将发布的版本之外的业务需求一定要确保不能混到release分支之内(避免由此引入一些不可控的系统缺陷)。成功的派生了release分支之后,develop分支就可以为“下一个版本”服务了。所谓的“下一个版本”是在当前即将发布的版本之后发布的版本。开发人员可以在此分支上修改BUG,进行提交、回滚等操作,但是与feature不同的是release分支是被多人操作的,不像feature,所以一定要小心避免冲突。当现在QA测试没有问题,便从release上发布上线,且经过一段时间的验证没有问题后合入master,并且删除release分支,其实根据release分支的特性我们可以使用Git Hook触发软件自动测试以及生产环境代码的自动更新工作。这些自动化操作将有利于减少新代码发布之后的一些事务性工作。

• 命名:

release/1.2.3 后面跟对应的版本号

• 流程:

同feature

Hotfix:

• 描述:

Hotfix叫热修复分支,除了是计划外创建的以外,hotfix分支与release分支十分相似,当已经发布的版本(Master上代码)遇到了异常情况或者发现了严重到必须立即修复的软件缺陷的时候,就需要从master分支上指定的tag版本拉取hotfix分支来组织代码的紧急修复工作。这样做的显而易见的好处是不会打断正在进行的develop分支的开发工作,能够让团队中负责新功能开发的人与负责代码紧急修复的人并行、独立的开展工作。

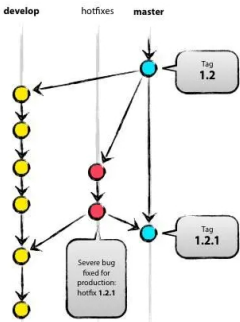
• 生成及销毁:

由Master上拉取,进行修复,负责修改BUG的同事可以进行提交及其它操作,后续的热修复测试也在此分支上进行。通过测试验证没问题后有一个人(通常为teamleader)**合并入Master分支,且同时也要合并入Develop分支**

• 命名:

Hotfix/1.2.3 后面跟对应的版本号

• 流程:



image

总结

Git Flow开发模型从源代码管理角度对通常意义上的软件开发活动进行了约束。应该说,为我们的软件开发提供了一个可供参考的管理模型。Git Flow开发模型让开发代码仓库保持整洁,让小组各个成员之间的协作更简单,从而在敏捷开发下开发团队在保持小步快跑的同时也能保持所开发

17成员之间协作相当简单，能够有双避免由于开发人员在同时代时相互影响而导致的效率低下和混乱。

所谓模型，在不同的开发团队，不同的文化，不同的项目背景情况下都有可能需要进行适当的裁剪或扩充。

效率工具

推荐 [sourceTree](#) 和 [gitk](#)(用免费版即可，不用充钱) 前者对gitsubmodel的支持不太好，不过目前介于我们没有实现组件化所以暂时可以无视；后者完美支持gitsubmodel，但是在拉取一些比较大的库的时候可能会卡死，前公司一个项目30G+ 会有卡死情况出现，后者界面炫酷一些 iOS的话Xcdoe自带git 也可以试试。

3人点赞 >

发布的文章

...

“小礼物走一走，来简书关注我”

赞赏支持

还没有人赞赏，支持一下



茉莉儿

欢迎来到我的网站<https://oliverqueen.cn/>

总资产15 (约1.43元) 共写了4.7W字 获得371个赞 共283个粉丝

关注

写下你的评论...

全部评论 0 只看作者 按时间倒序 按时间正序

被以下专题收入，发现更多相似内容

Hexo-NexT

推荐阅读

更多精彩内容 >

GIT开发模型文档整理

原文推荐: A successful Git branching model 这个文章讲的是Git分支模型的原理及...

小菜鸟_Sonya

阅读 713 评论 0 赞 0

Git开发流程整理

Git分支管理 master:主分支，当前分支上的代码随时可以直接发布，并且只能通过Pull Request从其他...

UEUEO

阅读 4,101 评论 4 赞 26

git学习笔记

学习资料pro gitgit - 简明指南Github官方帮助文档Git Community Book 中文版参考...

合肥鹏皮

阅读 10,239 评论 1 赞 17

git分支管理规范

Git 仓库申请流程 1. 开发主管向Git 管理员提交Git 仓库申请【邮件：发送给Git 管理员，抄送给项目经...

蛋包蛋天虎

阅读 500 评论 0 赞 0

如何做好幼儿园多元化经营？

幼儿园多元化经营的含义 1、战略性含义： 幼儿园的多元化经营，是指围绕“幼儿的发展、教师的成长、园所的质量”，制定...

园长头条

阅读 103 评论 0 赞 2

