# Android例会记录

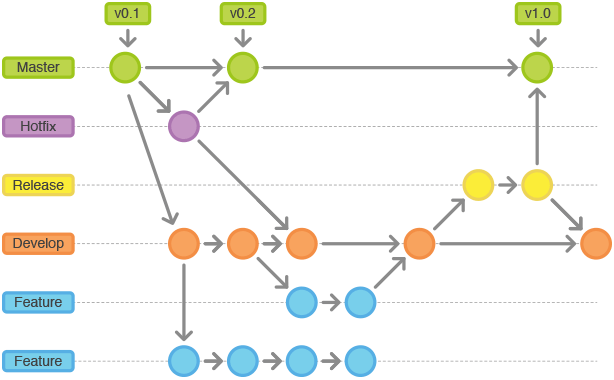
编写日期：18 - 3 -8

编写人： Vane

会议议程

1. Git 工作流
2. Restful API

Git 工作流 – Gitflow



1. 分支说明

* **Master** - 正式上线环境。为了保持该分支提交的信息的简洁性，在每次merge到该分支时，都应该合并commit，只保留正式发布的版本并打上tag。如上图所示，master分支 v0.1 到 v1.0之间，除了存在hotfix 信息，不应该存在开发过程中提交的信息。在每一个正式版发布时，应当打上对应的tag 标记。
* **Hotfix** – 线上环境bug修复分支
* **Release** – 预上线环境。当项目完成达到上线要求时(此前应该走一波测试)，切换预上线域名(不存在则使用测试环境)，再走一遍回归测试，有bug直接提交到Release分支，最终确认上线提交到master前再切换为生产环境域名。上线完成后，应该让Develop分支merge Release分支，并让Release分支Rebase Develop分支。
* **Develop** – 测试环境或开发环境。当完成某一Feature时，应当将代码merge到该分支，允许存在bug，但流程应该跑通。
* **Feature** – 特性功能分支。目前我们不使用该分支，取而代之的是个人开发分支，如Vane，Fred..分支。要求每个开发者应该每天至少进行一次commit，完成某一功能时merge到Develop分支。

1. Code Review

遏制开发者随意push代码至重要分支而导致分支被污染、分支信息混乱。项目负责人应该将master、Develop等重要分支设成保护分支，各开发者完成某一特性功能模块时，提交Merge request，项目负责人适时审查代码并处理各成员的Merge request。

1. 视项目复杂度选用工作流

Gitflow工作流的优点固然明显，但小项目使用Gitflow工作流就牛刀杀鸡了，且增加开发成本。所以简单的项目，我们更推荐使用集中式工作流或功能开发工作流。

Restful API

* 了解Restful API (参考：http://www.ruanyifeng.com/blog/2014/05/restful\_api)
* 思考Restful API 我们网络层该如何封装，可以怎么优化