**一、当前分支管理方案**

当前金穗慧生活后台代码分支管理方案为沿用之前多飞在分行大楼开发时的方案：

1.master分支作为开发分支&测试分支，所有人在此分支上进行开发，开发完成后直接推送至master分支，并发布测试环境测试；

2.测试通过后，由开发人员手动使用文本比对工具将master分支上的代码向release分支复制粘贴合并后提交推送至远端代码库，测试人员再在发布用的PC上从release分支拉取最新代码、替换配置文件、打包、发布。

该方案在实际操作过程中暴露出了诸多问题：

1．手工复制粘贴合并代码极其容易出现错误，从而导致“测试环境OK，上了生产就出错”的情况频出。

2.所有人在同一分支上开发，容易出现代码冲突的情形，给开发工作带来额外负担。

3.有些开发人员在需求紧急或者仅仅是图方便的情况下，直接在release分支上编码、提交、推送，导致master和release越行越远，代码差异越来越大，测试环境最终将不能作为生产环境的参照版本。

**二、新的分支管理方案**

1.基于release分支check out出一个develop分支，develop分支作为测试分支（暂定）。

2.所有开发人员基于develop分支check out出自己的开发分支，所有人在自 己的开发分支上进行代码编写；自己的开发分支命名规则为："branch-"+姓名拼音首字母小写，例如：branch-fyz

3．开发人员每次接收到开发任务后，切记首先将最新的develop分支代码merge到自己的开发分支（例：branch-fyz），代码编写完成后，先push到自己的开发分支，再merge到develop分支，待测试人员测试完成后再由develop分支merge到release分支投产，每次投产需打tag。

4.原master分支，因很多开发一半的代码未合并至release分支，不能确定后期是否有继续开发的可能，故将master分支做“冷冻”处理，所有人员请勿在此分支上提交代码；后期若有搁置的任务重新启动开发的，可从master分支上拷贝以前写的代码。

5.如遇到线上bug需要紧急发布版本，则由开发人员基于release分支直接check out出一个hotfix分支，在此分支上开发完成后merge到release分支和develop分支，待投产验证通过后将hotfix分支删除。

下图为开发人员每次开发任务的git使用周期：

