# 版本管理规范

# 目录

## 背景和目的

为了规范特性和发布管理，建立版本管理制度，保障版本可控，稳定迭代。版本管理的目标一是规范研发过程;二是保证版本可用;三是支撑特性研发。

## 版本管理工具

版本管理采用Git工具，版本要求在2.15+。

## **版本命名规范**

采用[Gitlab flow](http://www.ruanyifeng.com/blog/2015/12/git-workflow.html)模式管理分支，建立如下几个分支：

master：始终保持最新代码，除release分支外项目结束后都需要合并到此分支，任何新分支都从此分支创建。

hotfix：bug修复分支，命名为hotfix-{yyyyMMdd}-V1.0.0,为目标版本

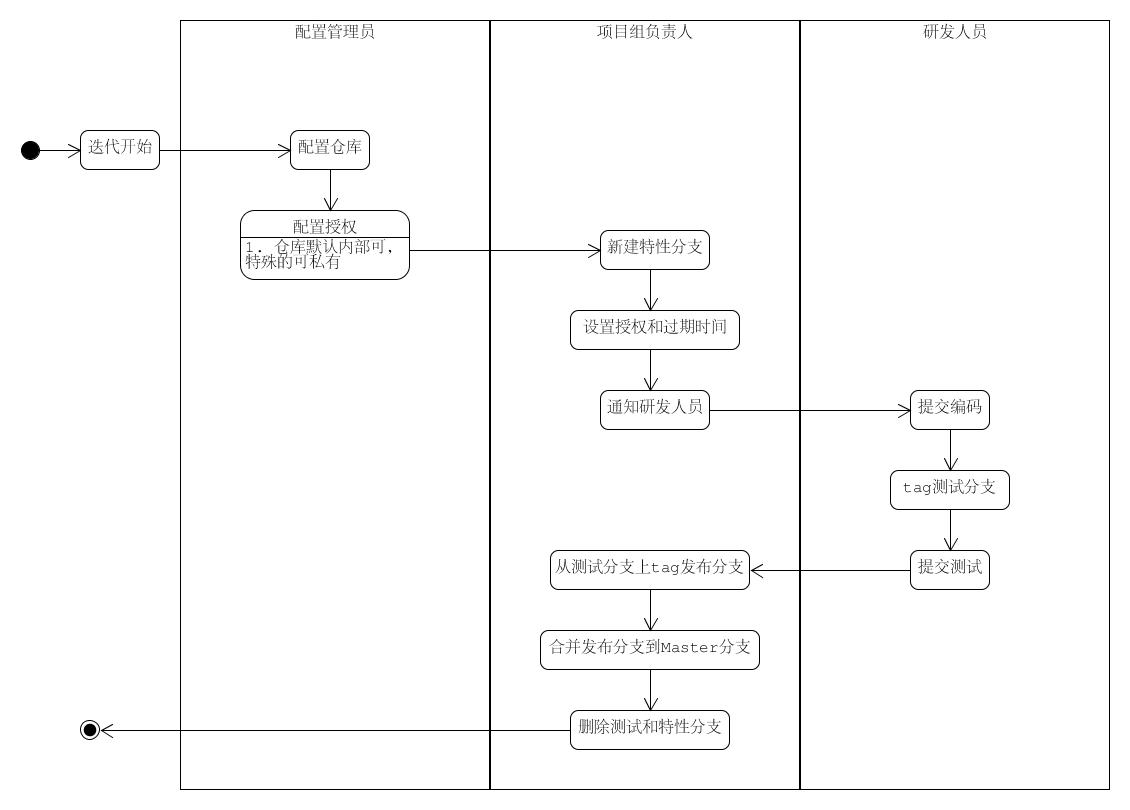
release：发布的分支，对应给外部的版本，命名为release-{yyyyMMdd}-V1.0.0

feature：特性分支，类似hotfix，命名为feature-{yyyyMMdd}-V1.0.0

test：测试分支，用于表示测试使用，命名为test-{yyyyMMdd}-V1.0.0-[buildNo]，buildNo表示构建号。

## **版本管理流程**

**版本管理需要配置管理员，研发人员，项目组负责人共同参与完成。具体流程如下：**

****

### 配置授权

配置管理员给项目负责人Maintainer权限，仓库默认内部可见，特殊情况下可设置私有。

项目负责人给每个研发人员Developer权限并设置有效期。测试分支给测试人员Reporter权限并设置有效期。

Release分支必须从test分支上新建。

### **分支记录**

特性分支和发布分支的关联关系须记录在仓库下的misc/vcs/{yyyy}.data文件中。