**编号：\_PRD/G04-0.1.0-2018-0.1.1\_\_**

**版本：\_\_ \_\_\_\_\_ 0.1.1\_\_\_\_\_\_\_\_\_**

**基于项目的案例教学系统**

****

**配**

**置**

**管**

**理**

**委托单位杨枨老师，侯宏仑老师**

**承办单位G04小组**

**修订历史记录**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **日期** | **版本** | **说明** | **作者** | **审批人** |
| 2018/10/19 | 0.1.0 | 完成配置管理初稿 | 周德阳 | 郦哲聪（PM） |
| 2018/11/15 | 0.1.1 | 修改封面及页码 | 刘乐威 | 郦哲聪（PM） |

目录

[1.版本命名策略 1](#_Toc530044851)

[1.1版本格式 1](#_Toc530044852)

[1.2版本更新 1](#_Toc530044853)

[1.3文档名称约束 1](#_Toc530044854)

[2.Git使用策略 1](#_Toc530044855)

[2.1基础知识 1](#_Toc530044856)

[2.2实际场景分析 2](#_Toc530044857)

# 1.版本命名策略

## 1.1版本格式

每一个文档的版本格式为[主版本号.子版本号.修正版本号。

示例：0.1.1

文档的初始版本为0.0.1。

## 1.2版本更新

当文件内容有了重大的变化或改进，主版本号加一。

当文档的内容有了模块的增加、补充等，子版本号加一。

当文档的内容有了小修改，如修正了纰漏等，修正版本号加一。

## 1.3文档名称约束

举例：PRD2018-G04-配置管理

# 2.Git使用策略

本次采用可视化工具github desktop工具进行同步操作与使用。

## 2.1基础知识

对关键字的理解：

Repository：仓库，可以理解为所有项目存放的地方。

Branch：分支是在同一时间在不同版本的存储库的工作方式。默认情况下，你的资料库中有一个名为一个分支master，其被认为是明确的分支。我们用树枝进行实验，他们承诺之前进行编辑master。

Merge branch：将B、C两个状态相对于A的改动合并到一起。注意，如果B、C对于A都只是增加内容，那他们可以轻易的合并到一起，但是如果在两个分支上改动了同一个地方，合并的就会出现代码冲突。 因为 git 不知道该听哪个分支的，所以只能报出冲突的位置，让开发者手动解决。

Push：将本地仓库的改动（包括你建立的本地分支）推送到远程仓库上，使其他组员也能看到你的修改。

pull：将远程仓库上的内容和本地仓库同步。

fetch：比对本地仓库和远程仓库的区别。

master：主分支，已经发布的内容。

备注：push之前请先fetch，看看远程仓库目前是不是最新版本，如果是的话先pull下来，再push，防止冲突。

## 2.2实际场景分析

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 场景 | 权限 | 操作分支 | 所在目录 | 上传注释示例 | 准备工作 |
| 协同编写某文档的0.1.版本，提交个人所负责的工作成果 | 项目组所有成员 | 默认为master，或配置管理员指定的其他分支 | 非受控文件 | [周德阳]提交《项目总体计划》[0.1.0]版本的1、引言部分 | 在提交前先pull，再进行修改/添加。 |
| 更新现有文件 | 项目组所有成员 | master | 受控文件对应阶段文件夹 | [郦哲聪]更新《可行性分析》[0.1.0]为[0.2.0] | 在提交前先pull，再进行修改/添加。 |
| 对配置管理系统中的文件命名进行整改 | 配置管理员 | master | 受控文件夹对应阶段文件夹 | [周德阳]整改文件命名 | 在提交前拉取远端的最新master，并以此为基础再提交 |
| 提交会议记录 | 会议记录员 | master | 受控文档对应阶段文件夹 | [冯一鸣]提交《会议纪要-10.06》 | 在提交前拉取远端的最新master，并以此为基础再提交。提交完成后还需要添加录音记录。 |

备注：配置管理员可以权限新建分支、合并分支。