****

**软件工程系列课程教学辅助网站**

|  |  |  |
| --- | --- | --- |
| 文件状态：  　[]草稿  　[√]正式发布  　[　]正在修改 | **文件标识：** | PRD-2017-G01-CM |
| **当前版本：** | 1.0.0 |
| **作者：** | 胡子阳，徐洁岑，何圳青，陈哲凡，吴苏琪 |
| **完成日期：** | 2018-1-12 |

**配置管理**

**Configuration Management**

**版 本 历 史**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **版本** | **作者** | **协助者** | **起止日期** | **备注** |
| 0.1.0 | 陈哲凡 | 胡子阳，徐洁岑，何圳青，陈哲凡，吴苏琪 | 2017/10/13-2017/10/13 | 起草 |
| 0.2.0 | 陈哲凡 | 何圳青，胡子阳 | 2017/10/14-2017/10/14 | 统一了版式 |
| 0.3.0 | 陈哲凡 | 陈哲凡 | 2017/10/30-2017/10/31 | 详细描述了提交文件的规范，具体内容为2.3、使用场景 |
| 0.3.1 | 陈哲凡 | 陈哲凡 | 2017/10/31-2017/10/31 | 增加了2.3对会议记录提交规范的描述 |
| 1.0.0 | 陈哲凡 | 陈哲凡 | 2017/1/12 | 发布 |

目录

[版 本 历 史 2](#_Toc495750551)

[1. 版本命名策略 3](#_Toc495750552)

[1.1. 版本格式 3](#_Toc495750553)

[1.2. 版本更新 4](#_Toc495750554)

[2. Git使用策略 4](#_Toc495750555)

[2.1. 基础知识 4](#_Toc495750556)

[2.2. 注意点 4](#_Toc495750557)

[2.3. 使用场景 5](#_Toc495750558)

# 版本命名策略

## 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0.1.0。

## 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。

# Git使用策略

## 基础知识

在使用之前，我希望每位组内成员都能明白git的基本用法与术语，在此，我对几个关键术语做出解释，如果不能理解，可以网上搜索资料或者问陈。

* 仓库：可以简单的理解为一个文件夹
* 多版本：如果一个仓库是多版本的，那么我们可以随时把它切换成某个时间段的某个样子，即不同版本。
* 分支：一个文件目前是A状态，甲将这个文件从A状态修改到了B状态，乙将这个文件从A状态修改为了C状态，那么从A这个时间点分叉出了两个不同版本（B、C）,即分支。
* 分支合并：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，如果对同一个部分有了修改操作，会造成“冲突”，需要人工合并，应该尽力避免这种情况。
* 远程仓库：即我们放在码市或者github的仓库，对于组员来说是共用的，上面的内容大多数应是可发行的版本（做完的）。
* 本地仓库：就是你自己电脑上从远程仓库克隆下来的文件夹，如果你只是在本地做了修改，是不会影响远程仓库的，其他组员是看不到你做了什么的，除非你push了改动。
* 远程分支：即远程仓库上不同的分支，所拥有的不同版本，对所有组员可用。
* 本地分支：你为自己在本地的仓库建立的分支，你可以选择是否push它，使它成为远程分支。
* 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同步到本地仓库上。
* fetch：可以检测出远程仓库对于你的本地仓库有哪些更新。
* master分支：主分支，上面的所有内容应保证是可用的、可发行的。

## 注意点

* 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。
* 对于push时，备注应该详细，比如对哪些文件的哪些部分做了何种修改，而不要笼统的说修改了某个文件

## 使用场景

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 场景 | 权限 | 操作分支 | 所在目录 | 上传注释示例 | 准备工作 |
| 提交个人作业 | 项目组所有成员 | master | 非受控文件/0x-组员名（如01-胡子阳） | [1-陈哲凡]提交个人作业《人月神话》读后感 | 在提交前拉取远端的最新master，并以此为基础再提交。 |
| 协同编写某文档的0.1.0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/0x-组员名（如01-胡子阳） | [2-陈哲凡]提交《项目总体计划》[v0.1.0]版本的1、引言部分 | 在提交前拉取远端的最新分支，并以此为基础再提交，并在文档后“加下划线自己的名字简写”如“PRD-2017-G01《项目总体计划》\_czf”。 |
| 提交由一个人负责的文件 | 项目组所有成员 | master | 受控文件对应阶段文件夹，如受控文档/01-项目可行性报告 | [3-吴苏琪]提交OBS图[v0.1.0] | 在提交前拉取远端的最新master，并以此为基础再提交。 |
| 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，如受控文档/01-项目可行性报告 | [4-陈哲凡]更新《可行性分析》[v0.1.0]为[v0.2.0] | 在提交前拉取远端的最新master，并以此为基础再提交 |
| 提交整合完的[v0.1.0]文档 | 配置管理员 | master | 受控文件对应阶段文件夹，如受控文档/01-项目可行性报告 | [5-陈哲凡]提交《可行性分析》[v0.1.0] | 在提交前拉取远端的最新master，并以此为基础再提交 |
| 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 | [6-陈哲凡]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\09-会议纪要 | [7-徐洁岑]提交《会议纪要-10.31》 | 在提交前拉取远端的最新master，并以此为基础再提交。每次提交(commit)不仅包括会议记录的会议文档，还必须更新《PRD-2017-G01-会议记录录音链接》 |

* 注：没有版本跟踪记录的文件（除了会议纪要），如GANTT图、OBS图，需在文件名上跟上版本号，如：PRD-2017-G01-GANTT-v0.1.0.mpp
* 只有配置管理员有权限新建分支、合并分支。