软件配置管理计划

**修订记录**

**\* 变化状态：C――创建，A——增加，M——修改，D——删除**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **版本编号** | **变化状态** | **变更内容和变更范围** | **日期** | **编写人** | **审核人** |
| 1.0 | C | 创建 | 2018/4/25 |  |  |
| 2.0 | A | 增加5.版本库命名规则、7.版本控制流程、8.封版版本管理流程 | 2018/9/19 |  |  |
| 2.1 | M | 修改 6.版本命名规则、7.版本控制流程 | 2018/12/1 |  |  |
| 2.2 | M | 修改 6.版本命名规则、7.版本控制流程、8.封版版本控制流程 | 2019/07/08 |  |  |

# 目的

编写此文档目的是为在项目的配置管理过程中标识软件、源代码、构建包等工作产品提供可遵循的标准。以保证各个环境（开发、测试）的独立性，避免相互影响。并确保版本的正确性、唯一性、可追溯性。

# 适用范围

本规范适用于xx公司研发的所有项目。

# 适用原则

1. 多个版本（测试版本、发布版本）；
2. 多次合并（分支开发完后要及时合并，以免大量合并时出现难以解决的冲突）。

# 术语与缩略语

大版本：功能、模块完整的版本。

小版本：日常的bug修复、小功能模块修改等的功能、模块不完整版本。

版本库：Gitlab上的源代码仓库。

版本号：标识软件产品的变化的一组编号。

版本包：由程序、文档等组成的可执行程序包。

# 版本库命名规则

## 5.1 版本库命名规则

源代码仓库和目录仅可使用大写字母、数字、下划线，不允许使用特殊字符和空格。

仓库名组成：项目名\_扩展\_项目类型

项目类型：

服务端——SERVER

Web前端——WEB

安卓端——ANDROID

IOS端——IOS

小程序端——MINI

如：SMART\_PLUG\_NB\_SERVER

## 5.2 版本库分支命名规则

源代码仓库的分支仅可允许版本库名+大版本号，不允许使用特殊字符和空格。

分支名组成：项目名\_扩展\_项目类型\_大版本号

如：SMART\_PLUG\_NB\_SERVER\_V2.0

## 5.3 版本库封版分支命名规则

源代码仓库的封版分支仅可允许版本库名+小版本号，不允许使用特殊字符和空格。

分支名组成：项目名\_扩展\_项目类型\_小版本号

如：SMART\_PLUG\_NB\_SERVER\_V1.0

## 5.4版本库Tag命名规则

源代码仓库Tag仅可允许版本库+TAG+小版本号，不允许使用特殊字符和空格。

分支名组成：项目名\_扩展\_项目类型\_TAG**\_**小版本号

如：SMART\_PLUG\_NB\_SERVER\_TAG\_V1.0.5

# 版本命名规则



## 版本号组成

主版本号.次版本号.修正版本号[.编译版本号/内部提测号]

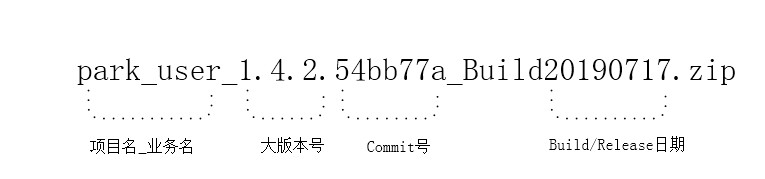
软件版本号有四部分组成，第一部分为主版本号，第二部分为次版本号，第三部分为修订版本号，第四部分为可选：编译版本号/内部提测号。

## 版本号修改规则

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **版本号级别** | **标识方式** | **作用** | **修改规则** | **修改决策对象** |
| 主版本号 | 由1开始的整型数值。 | 表示功能模块的的变动，比如增加多个模块或者整体架构发生变化。 | 主版本号加 1，次版本号及修正版本号复位为0。 | 项目组 |
| 次版本号 | 由0至99的两位整型数值。 | 表示功能的增加或变化，比如增加了对权限控制、增加自定义视图等功能。 | 主版本号不变，次版本号加 1，修正版本号复位为 0。 | 项目组 |
| 修正版本号 | 由0至99的两位整型数值。 | 表示Bug 修复或是一些小的变动，要经常发布修订版，时间间隔不限，修复一个严重的bug即可发布一个修订版。 | 主版本号和次版本号都不变，修正版本号加 1。 | 项目经理 |
| 编译版本号/  内部提测号 | Git前七位commit ID | 标识源代码版本号 | Git前七位commit ID | 开发人员 |

## 版本包命名

软件版本包分为：提测版本包和发布版本包。



项目名称/简称\_版本号\_Build/Release日期（YYYYMMDD格式）

版本包名称由三部分组成：第一部分为项目名和业务名；第二部分为版本号；第三部分为Build/Release日期。

第三部分Build为提测，Release为发布。

**提测、发布版本包命名举例：**

开发人员提测前，确定上一次提测版本号。

提测时，根据版本号修改规则并在上一次提测版本号基础上确定本次提测版本号的前三部分（首次提测版本号为1.0.0），第四部分用commit ID标记；发布时，将Build改为Release。

开发发现上一次提测版本号为：

angle\_detection\_1.0.0.e3495f3\_Build20180315.zip

1. 若本次提测主要为Bug 修复或是一些小的变动，commit ID为1511de，则本次提测时版本号在上一次版本号的基础上，主版本号和次版本号都不变，修正版本号加 1（项目经理决定）。若此版本可发布，发布时命名为：

angle\_detection\_1.0.1.1511de3\_Build20180419.zip

1. 若本次提测主要为功能的增加或变化，比如增加了对权限控制、增加自定义视图等功能，commit ID为1511de，则本次提测时版本号在上一次版本号的基础上，主版本号不变，次版本号加 1，修正版本号复位为 0（项目组决定）。若此版本可发布，发布时命名为：

angle\_detection\_1.1.0.1511de3\_Build20180419.zip

1. 若本次提测主要功能模块的的变动，比如增加多个模块或者整体架构发生变化，commit ID为1511de，则本次提测时版本号在上一次版本号的基础上，主版本号加 1，次版本号及修正版本号复位为0。（项目组决定）。若此版本可发布，发布时命名为：

angle\_detection\_2.0.0.1511de3\_Build20180419.zip

## 版本包格式要求

Windows用.zip， Linux用.tar.gz。

注：若程序包为.tar.gz格式，而整个版本包内又含有文档等需Windows下查看内容，则需要将文档等与.tar.gz 包一块再用zip压缩，即内层程序包用.tar.gz 格式，最外层用zip格式。

## 运维要求版本包格式

以下命名规则适用于测试环境与生产环境

1. 后端zip包，命名构成为：

项目名称\_服务名称\_version号\_commit号\_build号.zip

例如：park\_onenet\_v1.4.1\_b7fbd6a\_build20190529.zip

1. 前端zip包，命名构成为（前端打包要求为全量包）:

项目名称\_web\_version号\_commit号\_build号.zip

例如：park\_web\_v1.4.2\_7c4d773\_build20190529.zip

1. 小程序zip包，命名构成为：

项目名称\_服务名称\_version号\_commit号\_build号.zip

例如：park\_mini\_v1.4.1\_b7fbd6a\_build20190529.zip

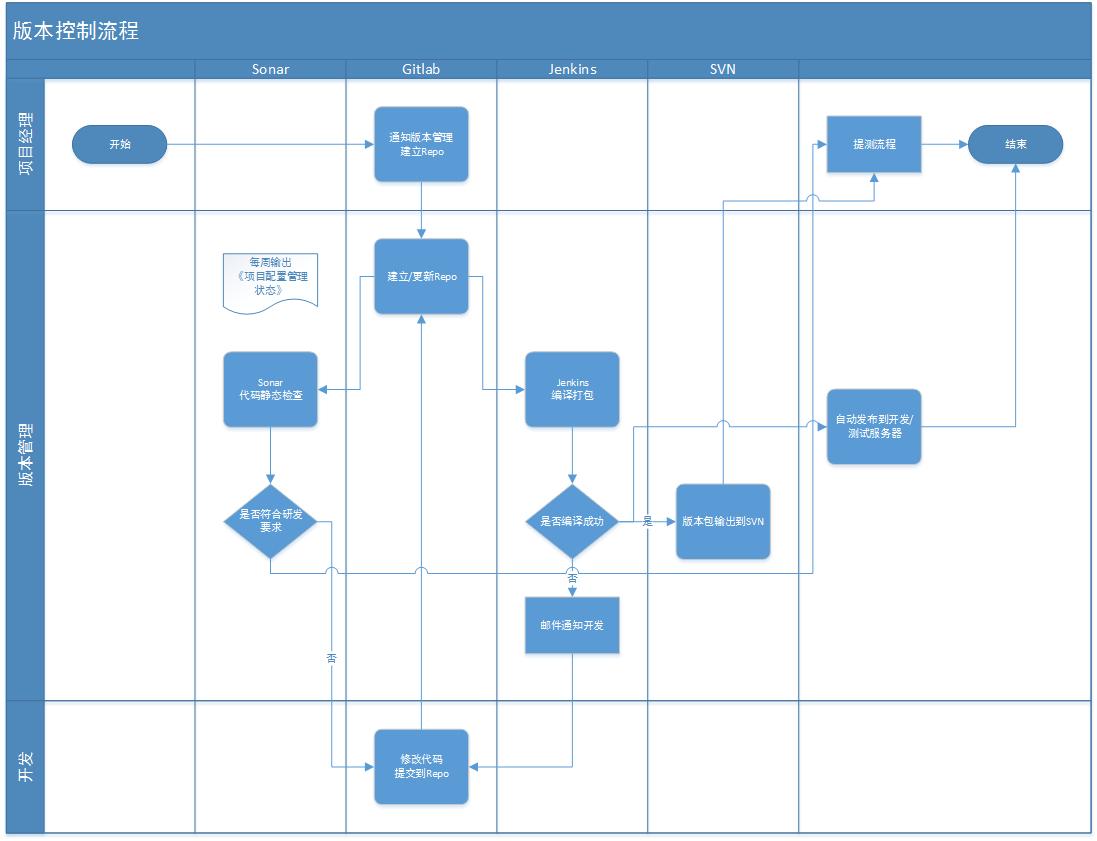
1. 移动端zip包，命名构成为：

项目名称\_服务名称\_version号（4位）\_commit号\_build号.zip

例如：park\_android\_v1.0.0.1\_b7fbd6a\_build20190529.zip

# 7．版本控制流程

## 7.1 版本控制流程示意图：



## 7.2 各角色职责和注意事项

**项目经理：**

1. 进入项目开发阶段后，确认版本库名称，命名需满足**5.1 版本库命名规则**，通知版本管理人员在Gitlab上新建项目。
2. 项目进入开发阶段的一周内，通知版本管理人员在Sonar上新建项目。
3. 项目进入开发阶段的两周内，通知版本管理人员在Jenkins新建项目编译打包。
4. 版本提测时检查下列项：

1) SVN上的版本包是否是Jenkins发布。（除现网问题外）

2) 版本名是否符合当前版本。

3) Sonar上的代码检查结果是否符合研发要求。BUG、漏洞、圈复杂度三项是目前研发要求必须修改项。

**版本管理人员：**

1. 根据项目经理需求在Gitlab上新建项目版本库。
2. 和开发沟通，在 Jenkins上新建Job，实现编译、打包、输出版本包到SVN。
3. 在Sonar平台上新建Job，进行代码静态检查。
4. 每周输出《在研项目代码检查报告》
5. IOS、MiNi（小程序）流程：开发把源码更新到Gitlab，在本地打包对应的版本包发给版本管理，由版本管理审核后更改版本包名并上传到SVN。
6. 版本管理人员需要进行开发环境部署能力联调时，需要获得项目经理的同意。需要进行测试环境部署能力联调时，需要获得测试经理的同意。

**开发人员**：

1. 若Jenkins编译报错，对问题定位修改后上传代码到版本库，要求问题当天解决。
2. 若Sonar检测结果不符合研发要求，对代码修改后上传代码到版本库。
3. Commit提交格式包含下列关键字：

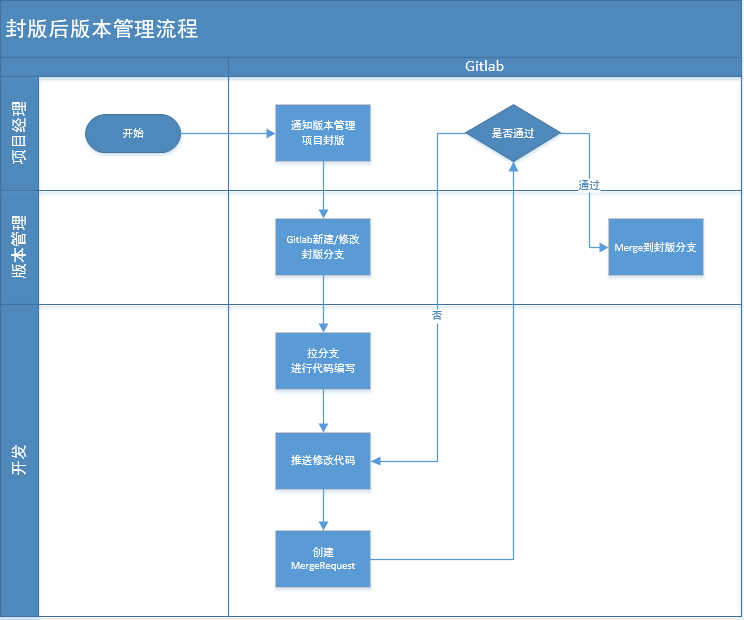
【功能问题】: 问题单号/需求单号，无需求单号时，写需求名称

【提交说明】: 问题的简要描述，需求的简要描述

【提交人】:提交人员名字

# 封版版本控制流程

## 8.1封版版本控制流程



## 8.2各角色职责和注意事项

**项目经理：**

1.版本封版后及时通知版本管理人员做封版操作。

2.收到Merge Request的邮件通知后，进行代码审查，对代码进行Merge。

**版本管理人员：**

1.拉取封版分支并命名。命名需满足**5.3 版本库封版分支命名规则**

**开发人员**：

1.拉封版分支代码进行代码编写。

2.修改代码并将其推送到单独的分支。

3.创建MergeRequst，并指定代码评审人员。具体参见**《Merge Request步骤》**