SVN 使用规范

文档版本： 1.2.1（试行）

适用范围： 基础技术部项目代码、技术文档

保密级别： 基础技术部内部

修订历史

版本状态	修订人	参与人	修订内容	修订日期
v1.0.0（初稿）	王德安		创建	2015-11-12
v1.1.0（草稿）	王德安		增加代码仓库结构	2015-11-18
v1.2.0（草稿）	方驰	王德安、刘永刚、江彩霞、陈绍邦	明确仓库使用规则	2015-11-26
v1.2.1（试行）	方驰		配图及完善描述	2015-11-30

一、规范作用

1) 方便对代码和文档的统一管理；

2) 方便对提交记录进行查阅；

3) 保证代码提交的及时性；

4) 避免并行版本开发时的冲突；

5) 方便对发布软件进行版本回溯。

二、配置说明

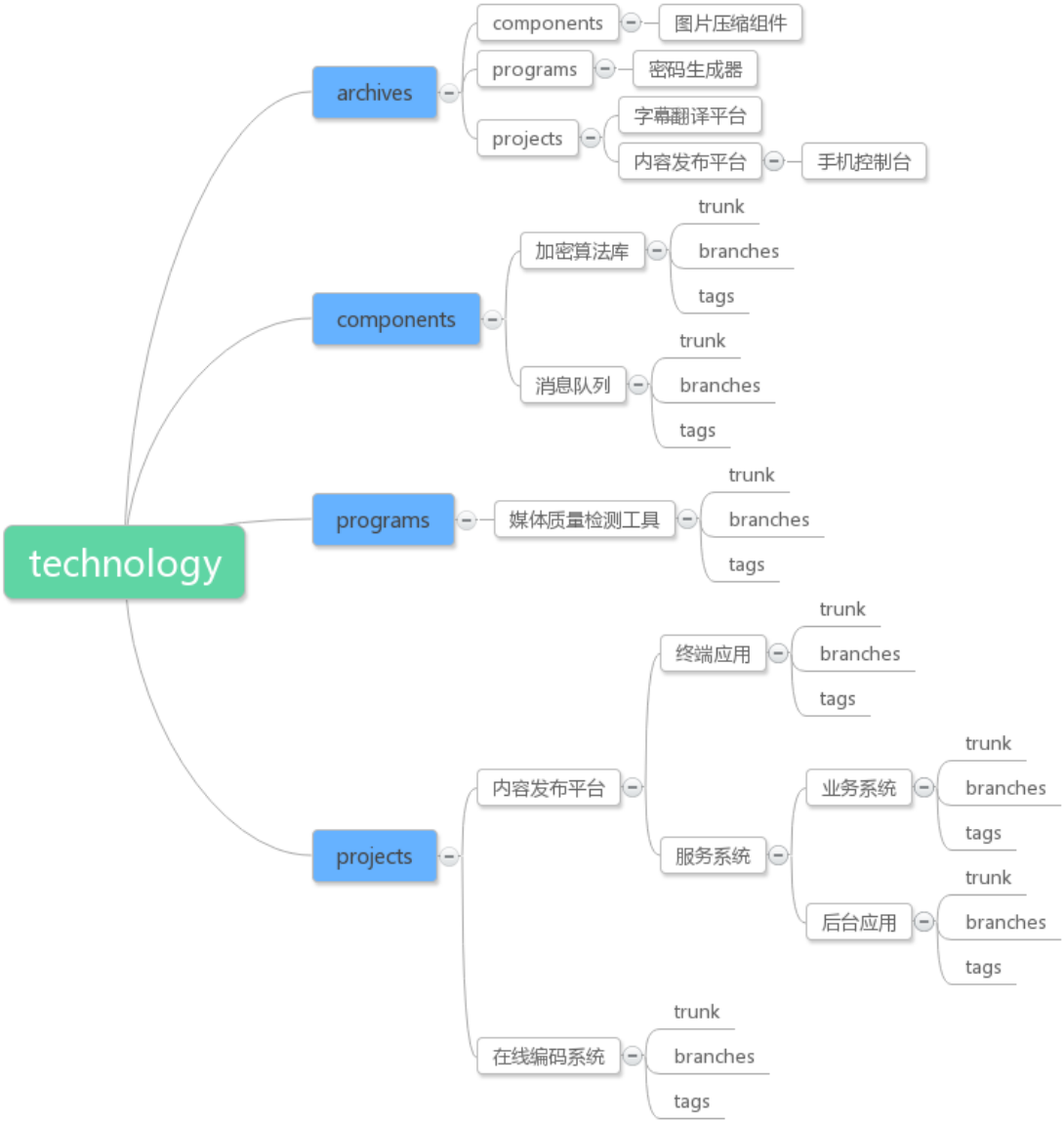
1) 部署一个独立的 SVN 库，仅用于项目开发；

2) 此 SVN 库仅限于开发人员使用；

3) 此 SVN 库仅用于存放项目相关的代码及文档。

三、库的结构

1. 结构示意图



2. 结构说明

1）SVN 库的名称为 technology，仅包含以下四个子目录：

- ✧ components：公共组件库的开发项目，提供其它项目开发使用
- ✧ projects：持续迭代或重点维护的系统型开发项目
- ✧ programs：工具型软件或非持续迭代的小型开发项目

- ✧ archives：对退役的开发项目进行归档

2）开发项目依据类型放置到 components、projects 或 programs 中；

3）开发项目可能包含了多个子项目，并对其中部分进行分类（比如示意图中，将“内容发布平台”的子项目“业务系统”和“后台应用”，放在“服务系统”的分类下）；

4）处于叶子节点的项目可以独立发布，发布时需要定义版本号，并在此 SVN 库中记录和管理。其所在的目录仅包含以下三个子目录：

- ✧ trunk：开发基线，无版本定义，包含最新的代码及文档
- ✧ branches：可以包含多个开发分支，无版本定义
- ✧ tags：已冻结开发的分支，版本已定义且可发布，但禁止直接提交更新

5）非叶子节点的项目其所在的目录，除了包含多个子项目的目录，还有可能存在一个仅用于存放综合性文档的 doc 目录（严禁存放代码）；

6）非叶子节点的项目在整体打包发布时，其版本定义不在此 SVN 库中记录和管理；

7）退役项目放入 archives 归档时，建议维持类似目录结构（见示意图）。

四、规则事项

1. 立项阶段

1）确定项目名称，然后在 SVN 中合适的位置创建项目的目录；

2）在项目的目录中创建 trunk、branches、tags 目录；

2. 设计开发阶段（迭代开发期间）

1）迭代开发周期中产生的项目文档和代码，及时提交到 trunk 中；

2）提交记录中准确描述每次提交的内容；

3）不同项目或分支的更新，分开进行提交。

3. 发布测试阶段

1）项目对外发布时，必须按照《软件版本命名规范》定义版本号；

2）创建 trunk 的分支到 tags 中作为版本标签，并以定义的版本号命名分支；

3）tags 下版本标签的描述必须清楚完整，包含主要需求和重要缺陷。

4. 设计开发阶段（维护项目的修订）

1）维护项目未进行新的迭代或发布时，trunk 内容与 tags 最新版本基本一致；

2）直接在 trunk 中修复缺陷并提交更新；

3）修复缺陷后的 trunk，定义新的版本号并在 tags 中创建版本标签。

5. 设计开发阶段（历史版本的修订）

1）项目进入新的迭代或修订更早的版本，trunk 内容已经脱离发布的版本；

2）从 tags 中选择对应需要修订的版本，创建分支到 branches 中；

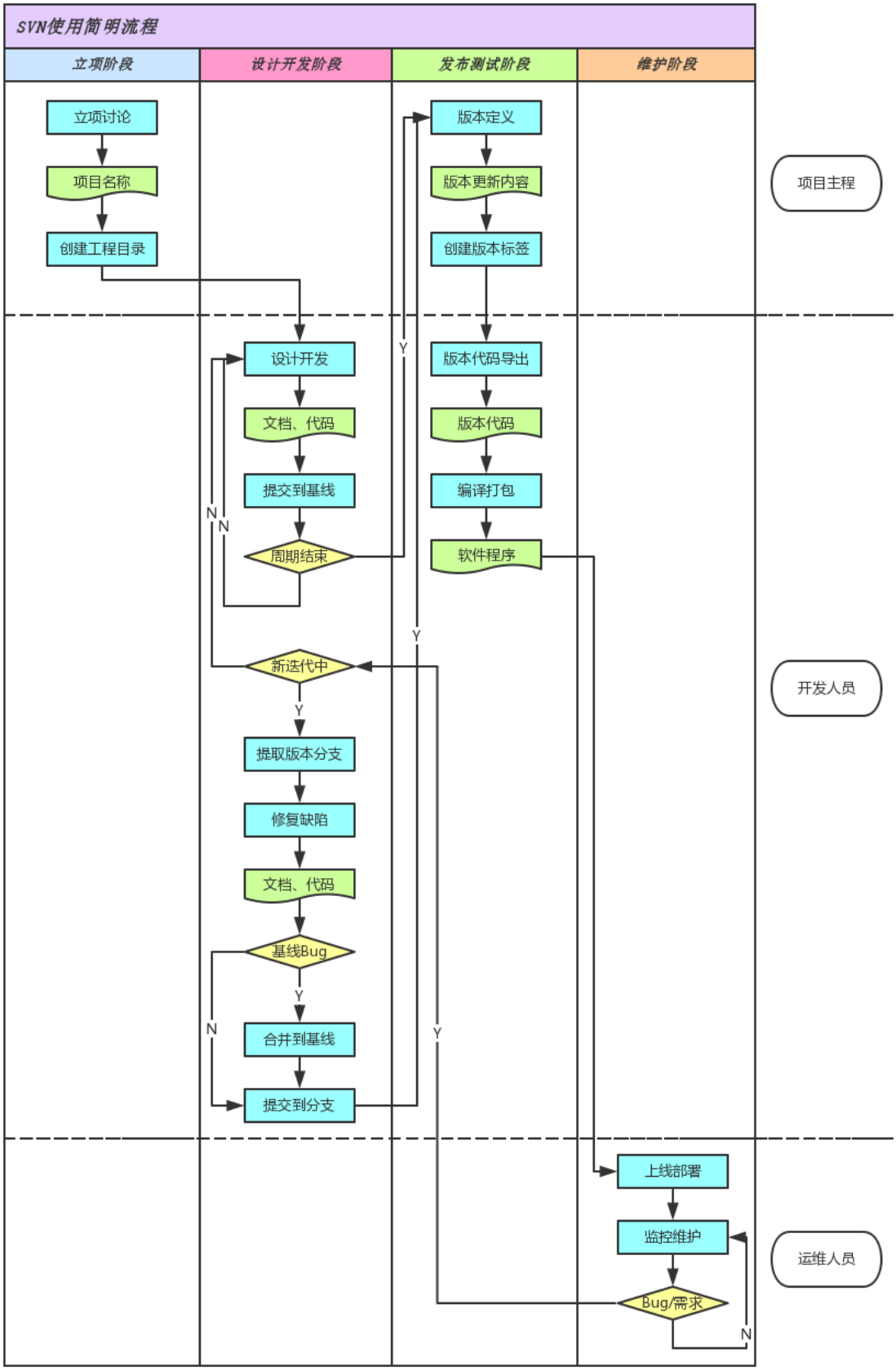
3）在 branches 的此开发分支中修复缺陷，并单独提交更新；

4）当缺陷在 trunk 中也存在时，同步修复 trunk 并单独提交更新；

5）修复缺陷后的 branches 分支，定义新的版本号并在 tags 中创建版本标签；

6）在适当的时候可以删除 branches 下多余的分支。

五、简明流程


SVN使用简明流程
立项阶段
设计开发阶段
发布测试阶段
维护阶段
立项讨论
项目名称
创建工程目录
版本定义
版本更新内容
创建版本标签
项目主程
设计开发
文档、代码
提交到基线
周期结束
版本代码导出
版本代码
编译打包
软件程序
Y
N
N
新迭代中
Y
提取版本分支
修复缺陷
文档、代码
基线Bug
Y
N
合并到基线
提交到分支
Y
Y
开发人员
上线部署
监控维护
Bug/需求
N
运维人员