**自研项目代码管理与规范 （初定版）**

**佳利达国际物流股份有限公司**

**目录**

[1、编写目的 3](#_Toc69480965)

[2、当前自研项目清单 3](#_Toc69480966)

[3、源码备份 3](#_Toc69480967)

[4、数据库备份 3](#_Toc69480968)

[5、版本控制 4](#_Toc69480969)

[5.1目的 4](#_Toc69480970)

[5.2具体实现 4](#_Toc69480971)

[6、代码质量管控 5](#_Toc69480972)

[6.1 编码规范原则 5](#_Toc69480973)

[6.2代码审查 5](#_Toc69480974)

1. 编写目的

为更好的管理好公司自研项目，保证源代码的安全性和有效性并提高代码开发效率，现对公司自研项目源码和版本进行统一标准化管理。

2、当前自研项目清单

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 项目名称 | 项目代码 | 源码地址 | Svn/Git |
| 1 | 金二报文转发CIP项目 | CIP\_ALL\_ONE | https://172.16.80.21/svn/JDLITSVN/  Projects/projects/ CIP\_ALL\_ONE | Svn |
| 2 | 关务辅助系统 | Entrance | https://172.16.80.21/svn/JDLITSVN/  Projects/projects/ Entrance | Svn |
| 3 | 毛利率报表项目 | supermodel | 外购项目，自研了一些存储过程配置，项目部署在172.16.1.91服务器上，无源码 | \_\_\_\_ |
| 4 | 柯美接口 | parcel | https://172.16.80.21/svn/JDLITSVN/  Projects/projects/ parcel | Svn |
| 5 | 关务数据中心CDC项目 | CustomsDataCenter | https://172.16.80.21/svn/JDLITSVN/  Projects/projects/ CustomsDataCenter | Svn |
| 6 | 固资系统 | AssetsManageSystem | https://github.com/JD-Link/gzgl.git | Git |
| 7 | 工单系统 | ITManagement | https://172.16.80.21/svn/JDLITSVN/  Projects/projects/ ITManagement | Svn |
| 8 | iCDC | iCDC | https://github.com/JD-Link/iCDC.git | Git |

注:目前自研项目源码大部分都在SVN服务器（172.16.80.21）上，固资系统和ICDC项目源码在Github上，后续会逐步迁移至Github。代码拉取和更新都会在Github上处理。

3、源码备份

目前源码都存储在公司内部服务器，有一些已经上传至Git，已经上线使用的软件源码建议服务器保留一份，Git上传一份作备份，存储硬盘考虑到硬件损坏的可能性，建议上传云服务器或者网盘，但一定要保证最新源码。

4、数据库备份

为保证系统的稳定运行，对于已经上线使用的软件进行定期（根据项目使用频率）数据备份，备份的sql文件以当前日期作为文件名称，服务器保留一份并同步更新至Github对应项目路径下。

正在研发中的项目可在项目上线后同步数据备份。

备份指令：$> mysqldump -u root -h host -p dbname > backdb.sql

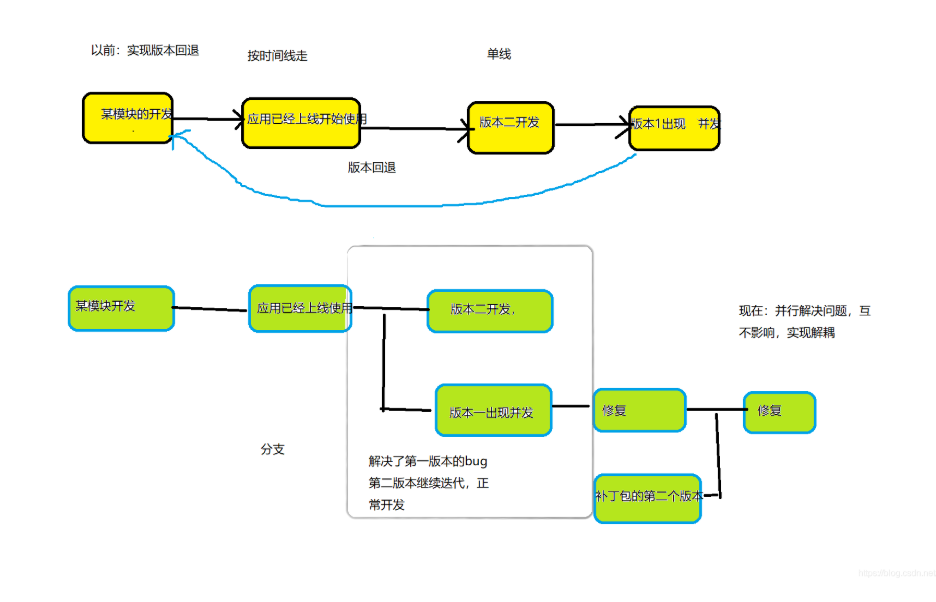
5、版本控制

5.1目的

版本控制的目的是实现开发团队并行开发、提高开发效率的基础。其目的在于对软件开发进程中文件或目录的发展过程提供有效的追踪手段，保证在需要时可回到旧的版本，避免文件的丢失、修改的丢失和相互覆盖，通过对版本库的访问控制避免未经授权的访问和修改，达到有效保护企业软件资产和知识产权的目的。

5.2具体实现

建议在Github进行版本控制，具体详见<https://blog.csdn.net/water_java/article/details/105205895>



本人也在研究Github版本控制，目前在摸索的过程中 ^\_^。

6、代码质量管控

6.1 编码规范原则

统一性：在程序中用同一种表示方法，同一个对象使用同一种表示方法。

达意：标识符能准确的表达出它所代表的意义，如果实在无法表达可使用拼音，并配有注释。

简洁：对于一个方法名或者接口名尽量缩短长度，在达意的基础上简介定义方法名，若实在无法简介可加注释弥补。

骆驼法则：除了包名，静态常量等特殊情况，大部分情况下标识符使用骆驼法则。

注释：1、在每个类前面必须加上类注释，必须要声明@date 、@author和类描述

2、在每个属性前面必须加上属性注释备注属性的意义

3、在每个方法前面必须加上方法注释，必须声明@param参数意义、@return返回类型和方法作用。

4、构造方法必须加注释，必须声明@param参数意义和方法作用。

5、方法内部注释，对于一些流程逻辑较复杂的代码段必须加上逻辑功能说明，例如第一步先做什么，下一步再做什么，给代码阅读人提供一些辅助性的思路。

注：具体编码规范详见 JAVA编码规范V1.0

6.2代码审查

Github可以review代码，建议在Git上进行代码审查，提交完成的代码须由相关人员review通过之后才能合并到项目里。Git管理员设置Branch Protection Rules 即可。

建议每天下班前30分钟上传代码，上传代码前自己先审查一遍，确实无误可上传，通知相关人员进行代码review,一方面看完成情况，另一方面看代码注释是否到位。