逻辑设计

全天不间断执行。。。每次执行完，休息（60分钟-运行时间）

1. 从数据库获取项目ID、URL等
2. 从数据库读取该项目ID之前的日期、versionID等
3. 根据之前日期，以及当前日期，通过svn log，获得xml，如果时间间隔长，以月为间隔，最早时间2016-7-1
4. 解析xml，获得需要检查的versionID
5. 通过svn diff versionId，获取diff文件（某些diff，可能要10分钟。。。恐怖）
6. 重新解析xml（因为这个解析很快），获取versionZ
7. 根据version，解析diff文件，写入svnLOC数据表

统计逻辑部分

1. 每天2点执行，以日为单位统计，统计每人在每项目的LOC
2. 检查数据库中是否已经统计，如果已经统计，则结束
3. 未统计，最后统计日期
4. 获取最后统计日期之后每天的数据，按天统计
5. 将数据读取到程序中，计算多少次数，多少提交，多少文件，多少行
6. 或者通过SQL计算，似乎逻辑更简单一些
7. 把结果写入数据库

# 项目基础表

此处假设有项目基础表信息，主要有项目ID、SVN路径这两个核心字段

|  |  |  |  |
| --- | --- | --- | --- |
| NO | 字段名 | 字段说明 | 其他 |
| 1 | PrjId | 项目ID |  |
| 2 | SvnUri | SVN路径 |  |
| 3 | SvnUser | svn用户名 | 非必须 |
| 4 | SvnPass | svn密码 | 非必须 |

# SVN代码行数据表

主要的表

注：只保持逻辑上是文本的文件，不处理非文本文件

|  |  |  |  |
| --- | --- | --- | --- |
| NO | 字段名 | 字段说明 | 其他 |
| 1 | ID | 流水ID |  |
| 2 | PrjId | 项目ID |  |
| 3 | SvnRevisionId | SVN**提交ID** |  |
| 4 | SvnUser | 用户 |  |
| 5 | SvnDate | 日期 | 如：2016-08-01 |
| 6 | SvnTime | 时间 | 如：2016-08-01 12:28:33 |
| 7 | SvnPath | 提交文件路径 | 如：/trunk/AplusPortal/WebRoot/jsp/account/dev/dev\_ModifyCharge.jsp |
| 8 | SvnFileType | 提交文件类型 | 如：jsp |
| 9 | SvnAction | 动作 | A\M\D等 |
| 10 | SvnLineAdd | 增加行数 | LOC-软件规模代码行(LOC, Line of Code) |
| 11 | SvnLineDel | 删除行数 |  |
| 11 | SvnKind | 文件类型 | File\Dir等，默认file |
| ~~12~~ | ~~SvnTextMods~~ | ~~是否文本模式~~ | ~~待用~~ |
| 13 | SvnPropMods | 是否配置模式 | 待用 |
|  | 。。。 |  |  |

# SVN代码行统计表

折算后的统计结果，是依据SVN代码行数据表计算出来的。

注：看是否需要，如果有，肯定对逻辑简单一些

|  |  |  |  |
| --- | --- | --- | --- |
| NO | 字段名 | 字段说明 | 其他 |
| 1 | ID | 流水ID |  |
| 2 | SvnUser | 用户 |  |
| 3 | SvnDate | 日期 | 如：2016-08-01 |
| 4 | CommitNum | 提交次数 |  |
| 4 | LineAdd | 增加的代码行 |  |
| 5 | LineDel | 删除的代码行 |  |
| 6 | LineModify | 修改的代码行 | 可能用不着，只有add和del |
| 7 | FileAdd | 增加的文件数 |  |
| 8 | FileDel | 删除的文件数 |  |
| 9 | FileModify | 修改的文件数 |  |
| 10 | AddTime | 统计时间 |  |
|  | 。。。 |  |  |

# 其他

SVN包含文件类型、SVN过滤文件类型、SVN文件类型折算配置，暂时保存在配置文件中，在开放过程中，根据工作量情况，再看要不要保存到数据库中进行配置