**GIT 代码提交规范**

|  |  |  |
| --- | --- | --- |
| **文件状态： [√] 草稿 [ ] 正在修改**  **[ ] 正式发布** | **部门** | **系统软件组** |
| **版本** | 0.1 |
| **作者** | **杨贵平** |
| **完成时间** | **20170317** |
| **审核** | **杨贵平** |
| **审核时间** | **20170315** |
| **密级状态：绝密( ) 秘密( ) 内部资料(√) 公开( )** | | |

**修改记录:**

|  |  |  |  |
| --- | --- | --- | --- |
| 版本 | 修订者 | 时间 | 说明 |
| 0.1 | 杨贵平 | 2017/3 / 17 | 基本版本, 只支持ADD/FIX/MOD的提交 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**一. 背景及问题:**

因为涉及到系统代码, project多, 维护的人也多, 修改历史也长. git commit 信息会很乱. 检查/查找很麻烦.

故特别对此作出要求. 要统一规范. 请大家严格按规范执行. 代码审批时, 也要检查提交信息是否符合规范.

**二. 思路和方法:**

commit 信息要完整, 包含标题及正文.

**标题要求:**

1. 开头用大写的关键字开头,

关键字有: ADD/FIX/MOD, 但要区分清楚:

ADD 代表新加的功能.

FIX 代表解决BUG.

MOD 其它的基本都可以用这个了.

这个要严格执行, 后续发布软件时生成修改历史就要用这个.代码审批时, 也要重点检查.

2. 关键字与正文之前用冒号分开

这个要求, 主要是方法阅读.

3. 标题可以用中文, 也可以用英文. 标题描述不要太长. 30个字母以内.

方便在gerrit中显示完整. 也方便阅读理解.

3. 结尾如果只有标题, 不要用符号, 如果有正文, 要加":"

方便在gerrit中显示完整. 也方便阅读理解.

**正文要求:**

1. 正文和标题之间要空一行. 方便阅读.

2. 正文在72个字符处换行. 每行不要太长.

3. 正文要说明是针对什么问题或为什么修改. 而不是如何修改.

如果修改的内容少, 可以只要标题.

如果内容比较多. 必需写正文. 说明详细情况.