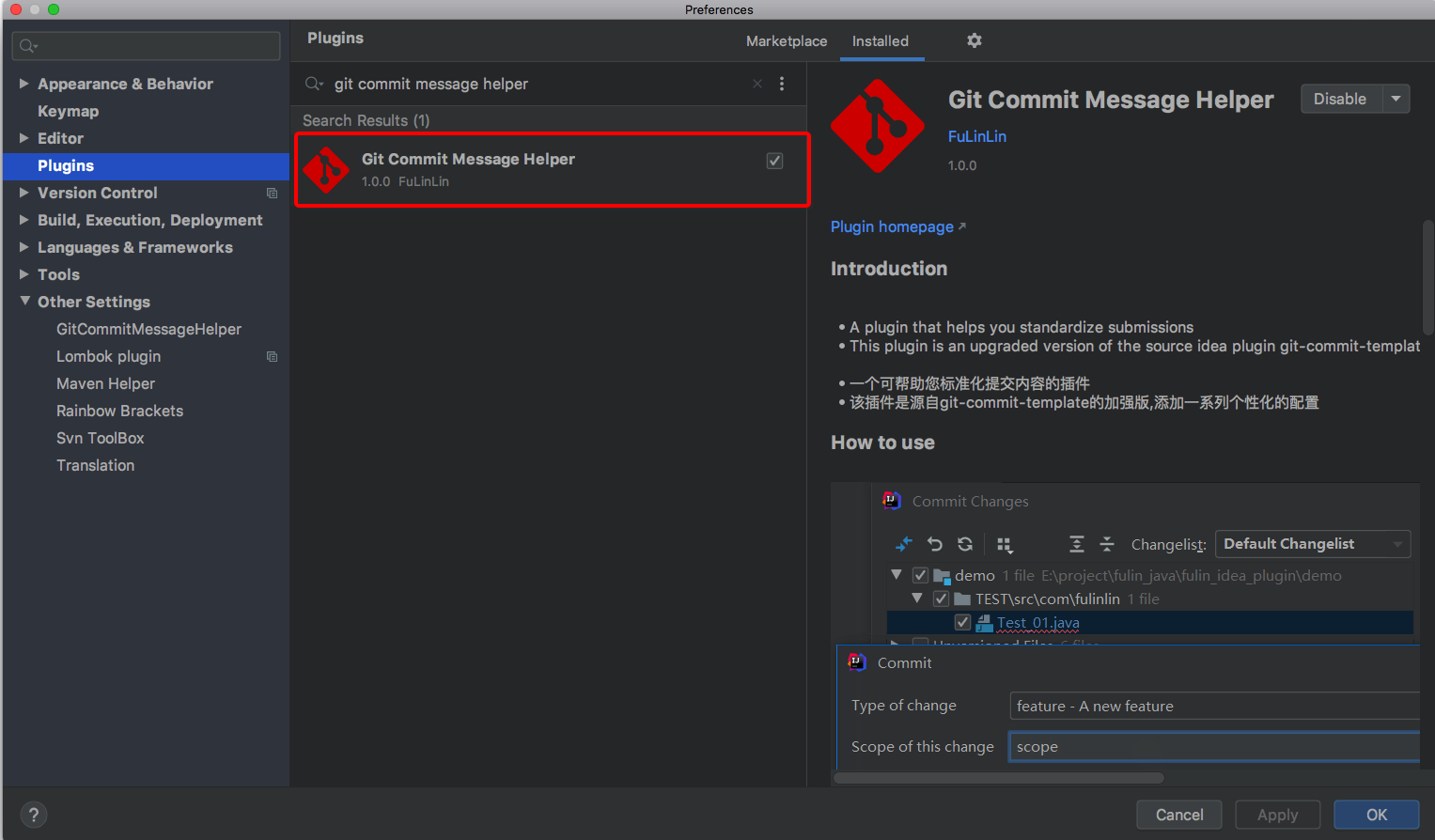
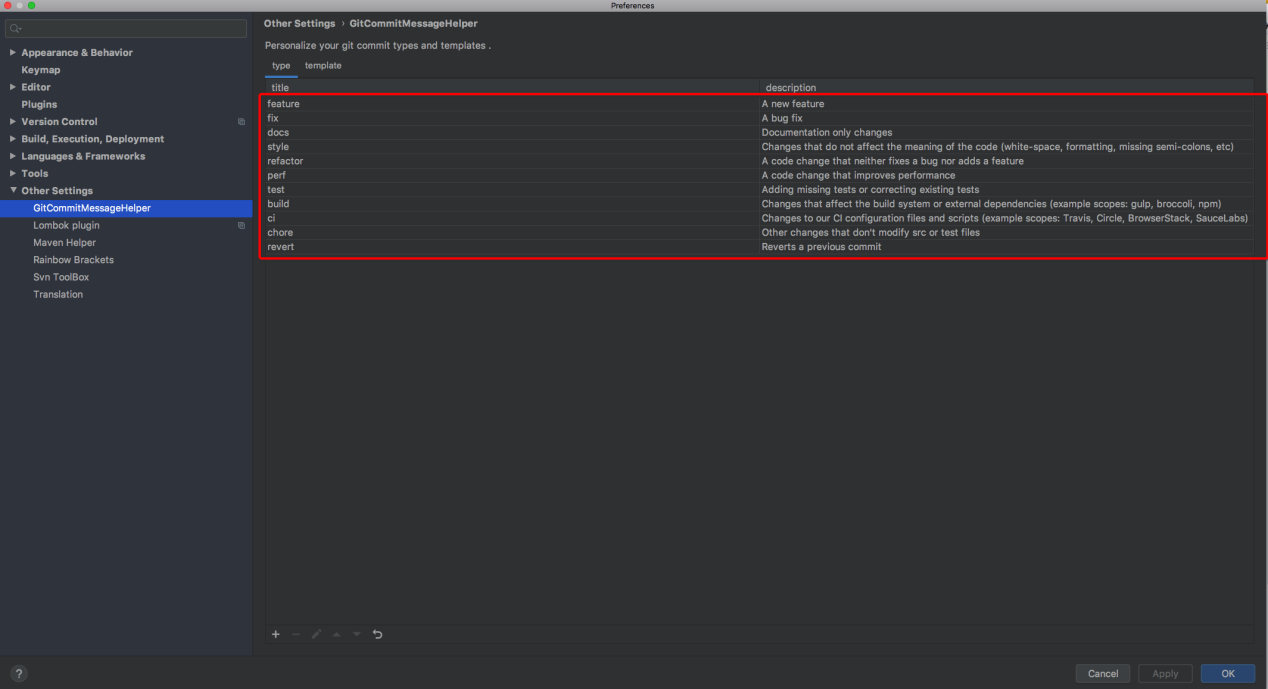
# MoNeed Git Message 标准化配置

## Idea 安装插件： Git Commit Message Helper



## 自定义Git Commit Message Helper 类型



### type用于说明 commit 的类别，只允许使用下面8个标识

feature：新功能（feature）

fix：修补bug

docs：文档（documentation）

style： 格式（不影响代码运行的变动,空格,格式化,等等）

refactor：重构（即不是新增功能，也不是修改bug的代码变动）

~~perf: 性能 (提高代码性能的改变)~~

test：增加测试或者修改测试

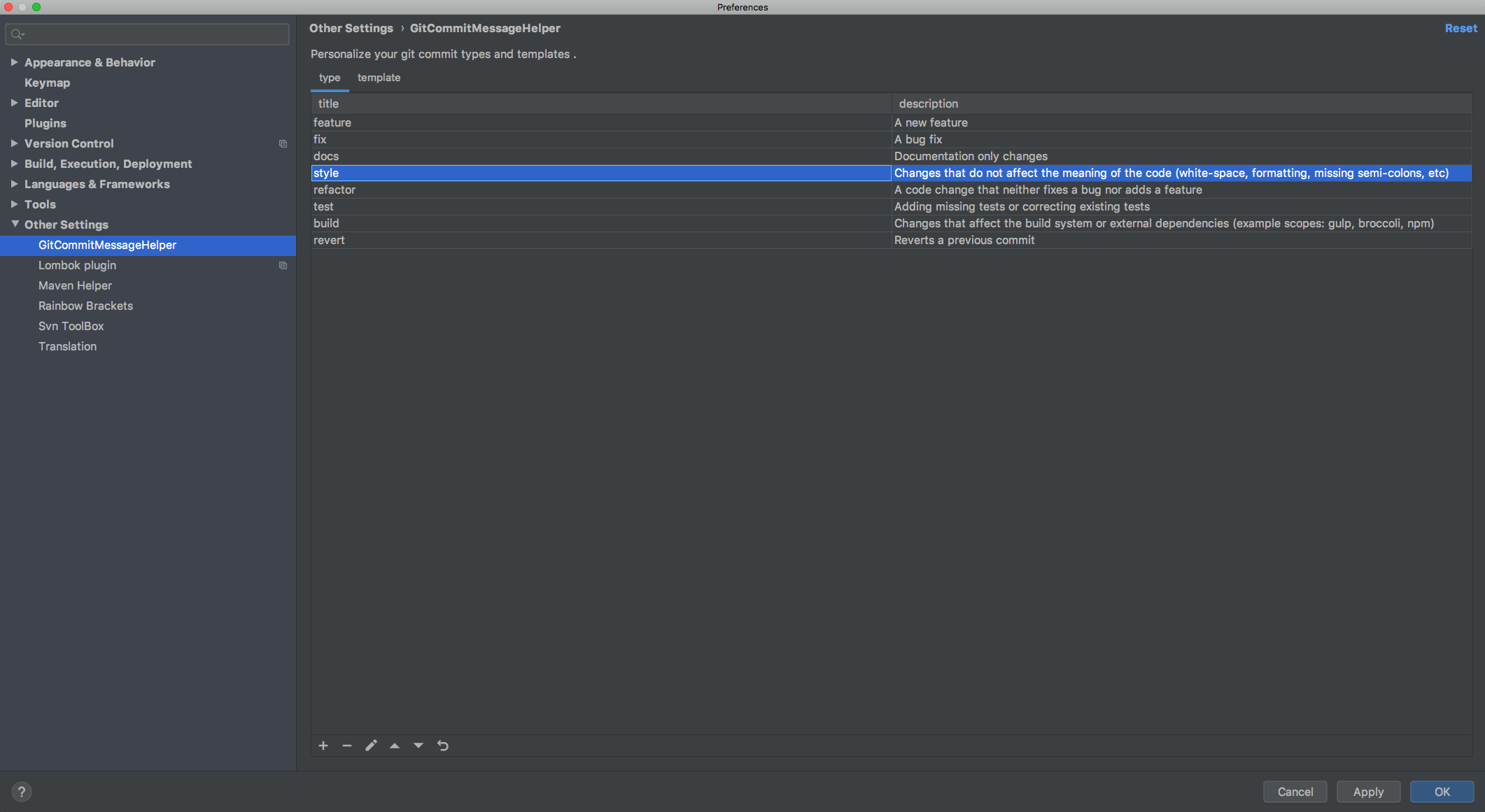
build: 影响构建系统或外部依赖项的更改(maven,gradle,npm 等等)

~~ci: 对CI配置文件和脚本的更改~~

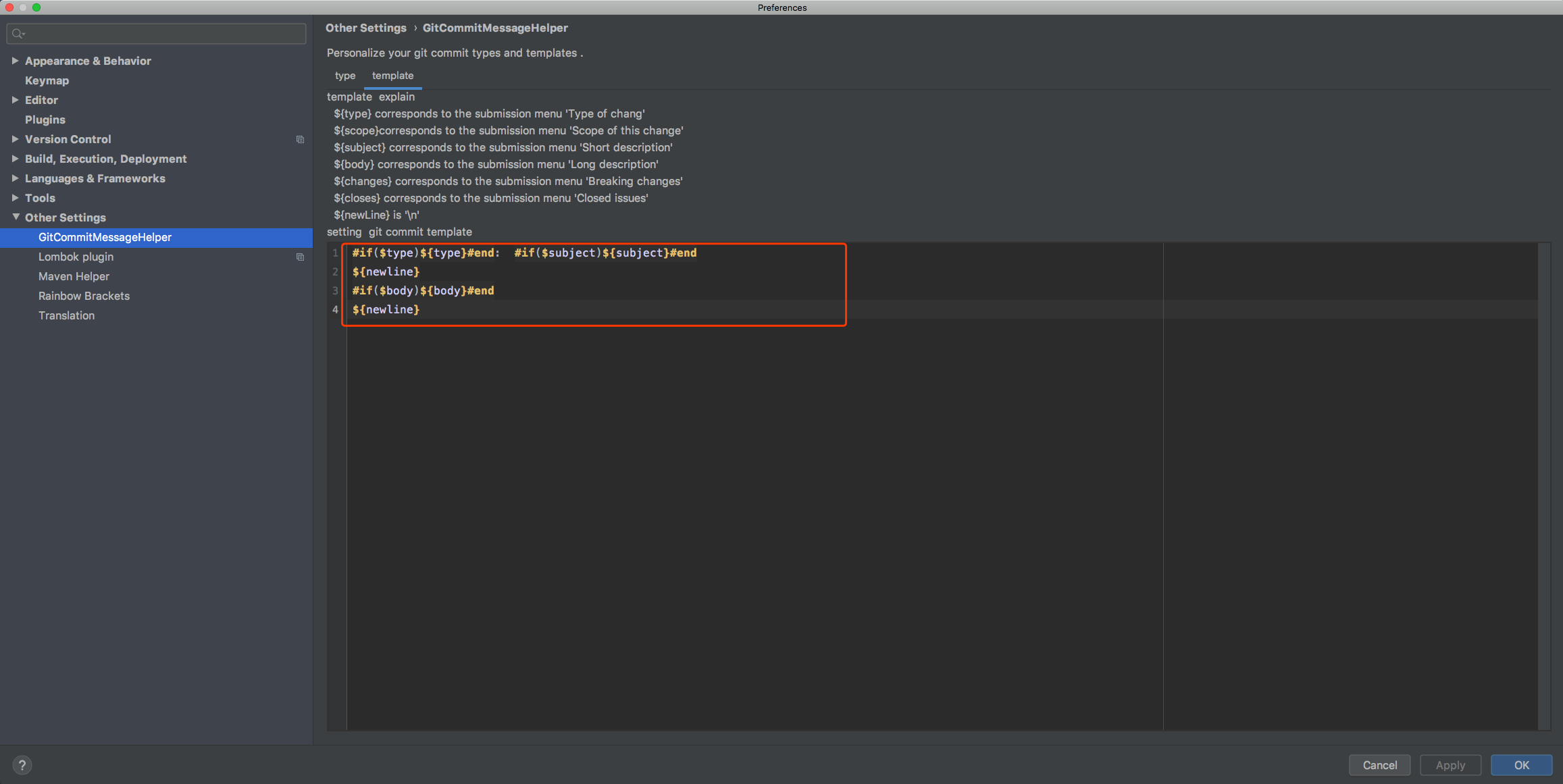
~~chore：对非 src 和 test 目录的修改~~

revert: Revert a commit

**我们只使用一下几类：手动删除 perf、ci、chore**



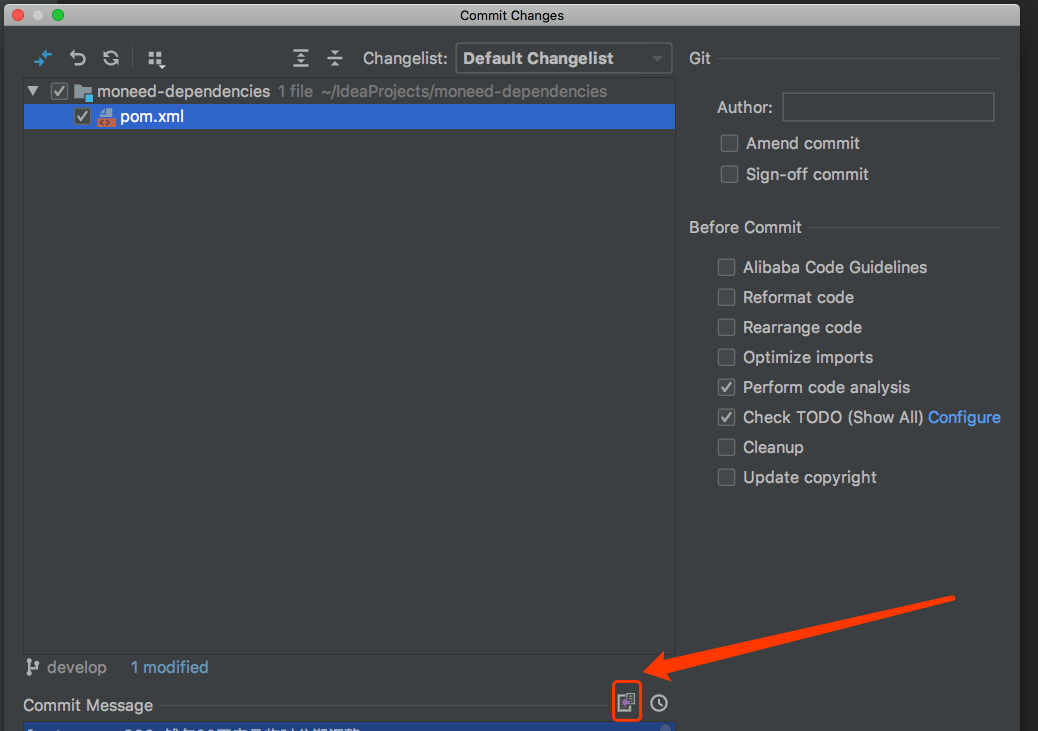
## 修改Git Commit Message Helper 模板



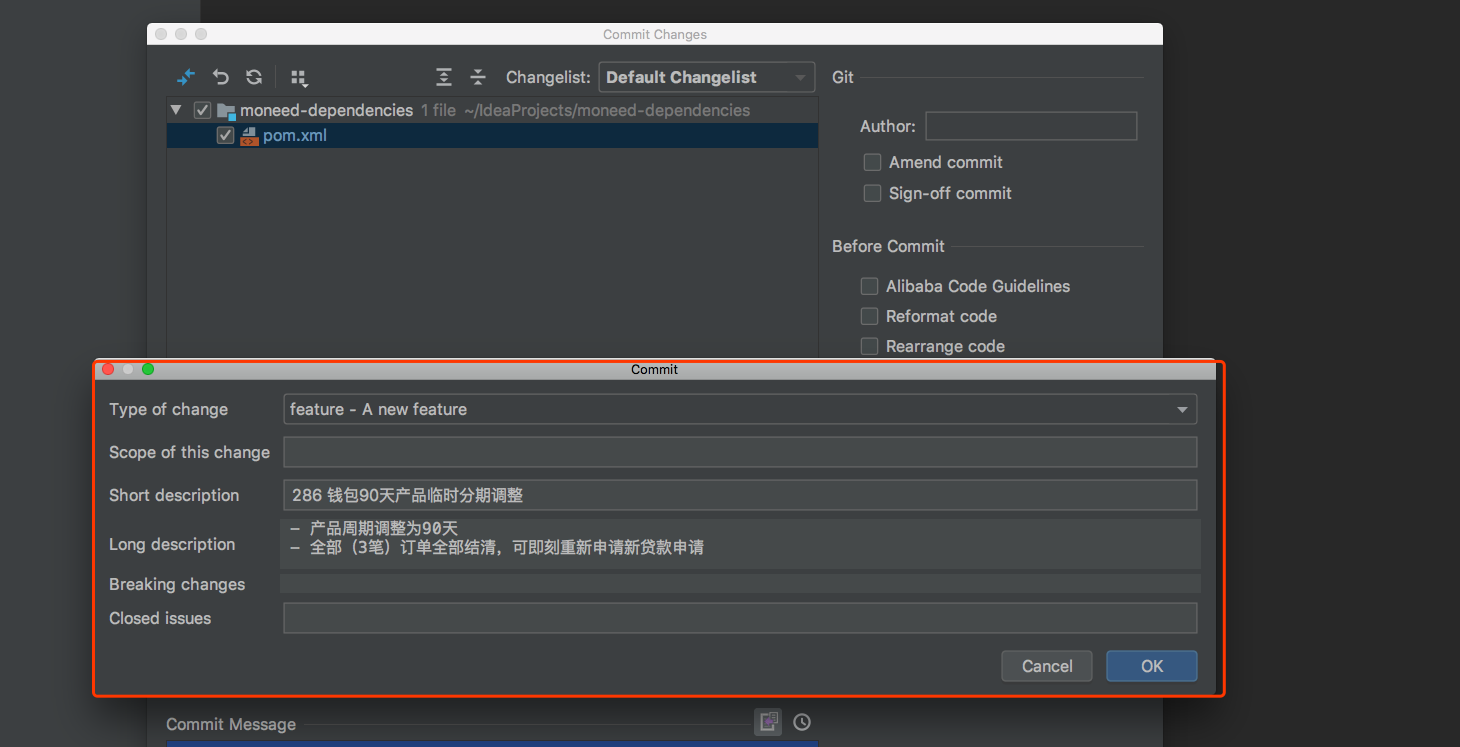
模板内容使用下面的内容替换：

|  |
| --- |
| #if($type)${type}#end: #if($subject)${subject}#end  ${newline}  #if($body)${body}#end  ${newline} |

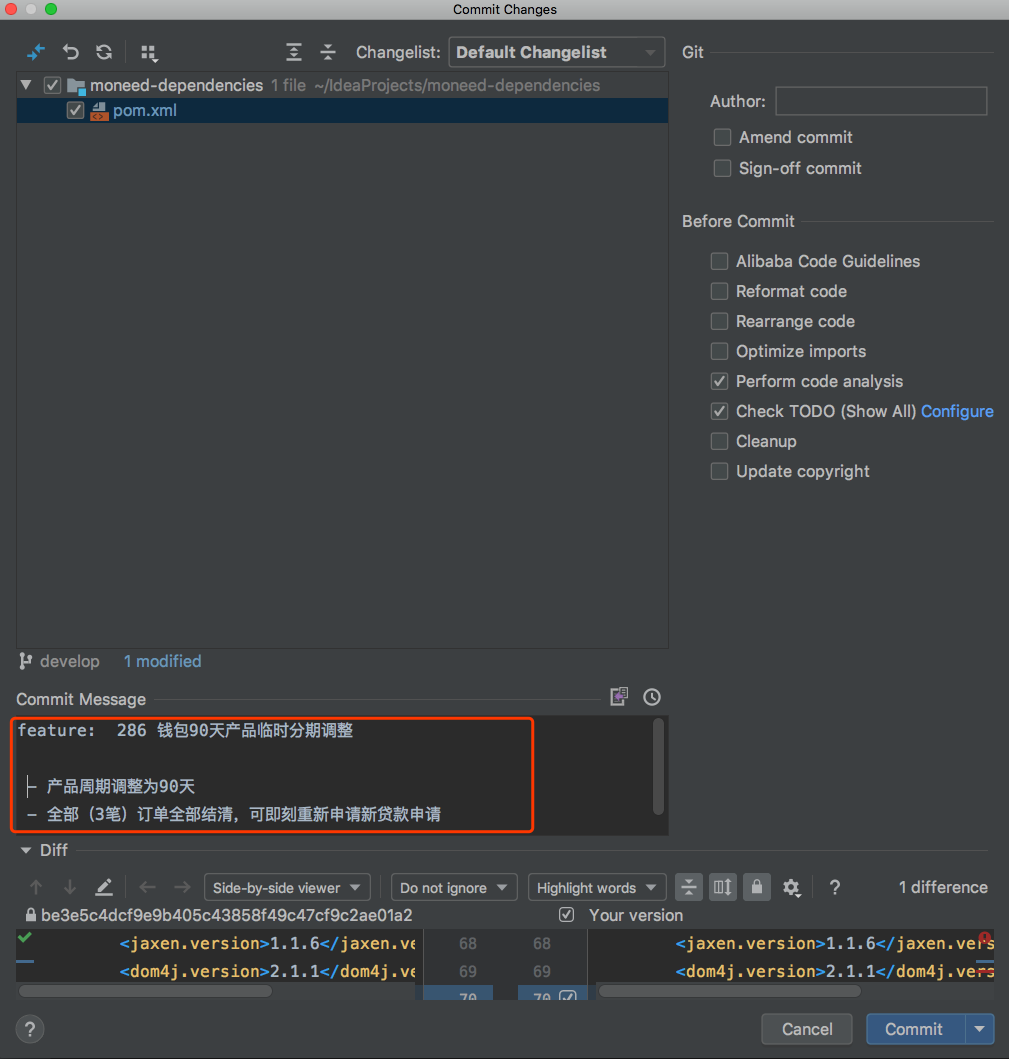
## Git 提交信息填写



**我们只需要填写 shot description 和 long description 即可。**



结果如下：



## 如果不想用插件， 那你就按照如下格式编写：

[task/fix/refactor] code message

例如：

task 256 放款失败增加操作次数

fix 125 （权限）内催管理--主管催收管理（在详情页点击返回按钮，页面报错）

refactor 000 本次重构功能的具体描述(必须是本次分支迭代相关的模块，不能重构无关的代码，重构代码不能改变原有代码的逻辑，因为会影响到测试工作和上线的流程和时间)

## 分类:

feature： 新功能（feature）

fix： 修补bug

refactor： 重构（即不是新增功能，也不是修改bug的代码变动）

docs： 文档（documentation）

style： 格式（不影响代码运行的变动,空格,格式化,等等）

test： 增加测试或者修改测试

build: 影响构建系统或外部依赖项的更改(maven,gradle,npm 等等)

revert: Revert a commit