分支：

Dev 开发分支，开发人员使用

Test 测试分支，测试人员使用

Master 线上分支，上线用

注：Test分支上不得合并未完成的功能和未fixed的bug，只有在完成某个功能或者fix某个bug之后，可以提交测试了才从dev合并到test，也就是说，test分支上的代码绝对是能运行，且按照常规来说没问题的代码（开发自测过，可以提测的代码）。

Commit文本规范：

格式：<type>( <id>): <subject>

Type：commit类型，有以下7种

1、feat：新功能（feature）

2、fix：修补bug

3、docs：文档（documentation）

4、style： 格式（不影响代码运行的变动）

5、refactor：重构（即不是新增功能，也不是修改bug的代码变动）

6、test：增加测试

7、chore：构建过程或辅助工具的变动

注：正常来说测试、后端人员只需要看1、2、5就可以了。

Id：项目管理工具上的任务或者bug id，多个用顿号相隔，只有1、2、5种类型才会有id。

Subject：commit 目的的简短描述，不超过50个字符。

例：

1. feat(#1321):增加\*\*模块
2. fix(#1234、#4635):解决文本超出及侧边栏自适应问题。

注：以上commit规范只适用于完成了某个功能开发或者fix了某个问题之后的commit，若是开发人员开发到一半没开发完的commit文本可以按照自己的习惯自由提交（但不得和以上七种冲突！建议形成自己一套提交规范，易于管理），也就是说除了前端人员之外的测试、后端开发只需要看以上七种类型的commit，其他可以自动忽略掉。