**开发规范建议**

1. **公共部分维护：在已经有了项目的基础上，接下来为了更加有效率的完成新项目的开发，需要对前端公共部分进行一个维护，包括脚手架和组件的维护（需要一个人专门维护这一块）**
2. 基础脚手架的开发（初始化项目公共的东西）
3. 公共组件的维护
4. **项目规范：为了项目更好的日后维护，需要有一套项目开发规范**
5. readme.md文件说明
6. 产品基本说明
7. 用到的通用组件
8. 页面逻辑需要用到的组件
9. 页面目录
10. 复杂，主要技术说明
11. 版本说明
12. js内容规范
13. 尽量用es6的语法去开发，统一用传统的方法定义或者统一用箭头函数
14. 引入js文件必须要有引用说明注释;
15. 参考JS规范开发
16. 用组件的方式去开发，达到一个解藕性，有利于代码的可读性和后期维护
17. 通用的方法要抽离出来，如果是单个页面的逻辑可以提取
18. css内容规范
19. 通用的scss要单独写在一个scss文件中：通用字体，通用主体颜色，通用布局
20. 有单独的mixin文件用于scss中的引用
21. 为了代码的可读性和日后的维护性，内容嵌套不要超过三层
22. scss注释要有组件化的思维，每一个大块有注释说明该
23. 统一用数据流的方式去开发,比如vuex,redux，不要在页面中存储太多公共的状态，不利于维护和代码可读性
24. 开发过程中尽量组件分离，通过解藕；更有利于后期维护
25. 文件命名规范
26. 文件夹的命名和文件的命名统一用驼峰式命名（eg:pageIndex）
27. 图片的名称规范：页面名称\_图标功能\_icon.png(eg:index\_header\_icon.png)
28. 图片统一保存为png
29. html标签的class定义风格统一用横线链接的方式去命名(eg: index-wrap)
30. 图片的引入方式统一使用定义变量的方式，不要在css里面引入图片（eg:import iconHeader from ‘../../assess/’）
31. **组件开发规范**
32. 头部统一说明组件名称（eg:<!—组件 – 头部菜单 -->）
33. 组件js头部需要有注释说明，组件的作用，需要的状态，和使用方法
34. 分离最小原则，尽量把一个页面分成几个小组件，单个组件之间维护自己的模版和状态，更有利于维护和数据流的清晰
35. vue开发避免使用mixin，mixin不利于后期维护和代码的清晰度，通过组件分离的方式去实现
36. **前端开发项目初期需要准备**
37. 初始化项目，需要集成如下功能
38. 打包，devserver
39. mock：需要在脚手架集成本地mock，让前端开发脱离接口
40. 项目需要用到到npm
41. reader.md文件说明初始化
42. 前端需要一份设计文档，设计文档包括（用xmind写）
43. 项目有多少页面，可以分成多少模块（根据UE）
44. 项目的交互逻辑(根据UE)
45. 项目需要用到的接口（这个是和接口设计一起进行，然后和接口对接）
46. 组件化说明（项目中可以组件话开发的东西，有利于后期抽离，公用）
47. 项目注意事项
48. **和后台合作建议**
49. **需要设计文档：方便前端开发和后期维护**
50. **设计评审和UI评审**