# AnyWork项目开发总结



## 一、前言

## 一）目的

1. 本次展示以我们之前开发的学习平台项目——AnyWork的整个流程为例，细数出我们在之前的项目合作有哪些不足与缺陷。
2. 通过AnyWork这一个项目，总结出一套工作室内合作的规范化流程，
   给工作室的其他成员以借鉴作用。

## 二）项目背景

AnyWork，是一个用于给学校师生在线练习与做作业的平台，原本产生于极客杯比赛，后被用于工作室项目，其后经历数次版本迭代，最终发展成为一个兼容网页与安卓端的在线考试平台。

## 三）开发流程预热

1. 需求分析
2. 接口设计
3. 功能实现
4. 测试修复
5. 部署上线

## 四）开发过程中的问题

1. 开发流程
2. 文档问题
3. 组内合作

# 二、开发流程

## 一）开发流程详细说明

第一步：确定项目组成员，总负责人，组负责人。

第二步：项目组成员一起讨论需求文档，由总负责人收集并编写需求初稿，经再次讨论确认无误 后发布需求文档v1.0版本。

第三步：项目组成员一起讨论接口文档，由总负责人确定函数原型初稿，各开发人员核实参数， 返回值，请求方式等无误后发布接口文档1.0.0版本.

第四步：各小组正式进入开发阶段，后台小组完成数据库文档交给总负责人保存（或由后台小组 负责人保存），并及时更新接口的接收和返回的数据格式示范案例。

第五步：开发过程中需要按时完成阶段性进度汇报，总负责人把握各个小组的开发进度。

第六步：测试与bug修复，开发过程中需要进行阶段性单元测试，整个项目完成后需要进行集成 测试、黑盒测试、白盒测试等。

第七步：开发完成，项目正式部署上线。

# 三、文档问题

## 一）前后对比

### a. 以前

1. 文档没有合并，没有汇总，各个模块负责人均持有自己的模块的文档，拥有绝对的读写权限，改动随意，造成文档混乱。
2. 没有版本迭代的意识，版本号随意添加或修改，经过几个版本的迭代后，功能需求开始变得模糊不清晰。
3. 文档内容的格式不统一、不规范，造成理解上的困难，对接效率低。

### b. 现在

1. 各模块文档需要合并，交由总负责人托管，只有总负责人有写的权限，其他人员只有读的权限。
2. 按层级目录保留下所有的迭代文档，每份文档都要标注版本号，并且需要指出和上一个版本的差异。
3. 规范化文档的内容和格式，按模版进行文档的内容编写。

## 二）文档格式

### a. 需求文档内容

1. 项目背景
2. 功能概述
3. 分模块图
    - 功能模块图
    - 用例图
4. 要点说明

### b. 接口文档内容

1. 更新内容
2. 数据统一格式
3. 功能接口模块

## 三）版本迭代

### a. 需求文档版本迭代

1. 版本号用两位数字表示，如v1.0版本。
2. 其中，第一位表示模块增删，第二位表示功能增删。
3. 每次更新，均需要重新出文档，不允许覆盖之前的内容。

### b. 接口文档版本迭代

1. 版本号用三位数字表示，如v1.0.0版本。
2. 其中，第一位表示模块增删，第二位表示接口增删，第三位表示接口修改。
3. 每次更新，若是第三位数字变动则直接覆盖，其余情况均需要重新出文档。

## 四）问题总结

1. 需求文档的编写规范。
2. 接口文档的编写规范。
3. 进行文档的版本迭代。
4. 文档编写的注意事项。

# 四、组内合作

## 一）版本控制使用

### a. 版本控制管理

1. 项目总负责人负责文档仓库，仅负责人有权利修改文档。
2. 小组负责人负责各自的代码仓库管理，添加开发人员为项目合作者。
3. 各组开发人员根据文档进行项目开发，最终整合到负责人的仓库中。

### b. 合作规范

1. 在小组负责人的远程仓库中添加开发人员作为合作者。
2. 合作者有适当的权限使用远程仓库。

### c. 分支规范

2个分支master和develop，他们的职责分别是：

- master：永远处在即将发布状态。
- develop：最新的开发状态。

## 二）移动组

### a. 确定组内负责人

1. 全权管理项目
2. 与其他小组定期汇报项目
3. 保持与组员的沟通
4. 跟进项目开发进度

### b. 模块分工

1. 确定项目需求
2. 确定分包和编码风格

### c. 前后端对接

1. 按接口文档进行编码，预留接口获取数据
2. 根据后台提供的接口开始获取数据
3. 通过postman测试对接过程出现的错误
4. 根据 postman提供的错误信息对接口进行改正处理

### d. 测试

1. 单元测试
2. 功能测试
3. App性能检测
4. 兼容性测试

## 三）前端组

### a. 确定组内负责人

1. 合理分配工作
2. 保持与组员的沟通
3. 跟进项目开发进度
4. 定期汇报项目

### b. 确定项目需求功能

在所有成员商讨确定需求文档后，前端所有成员需要清晰的了解整个项目的功能和交互过程，并且需要再开发前互相确认各自理解的功能和模块是否有偏差。

### c. 与设计师确定页面

1. 整体流程
2. 每个页面有什么功能，该功能需要那些组件
3. 需要如何交互
4. 那部分需要切图
5. 页面优先级

### d. 分配任务

1. 根据使用技术框架来决定
2. 确定页面跳转之间的依赖关系
3. 确定公共模块组间开发

### e. 开发阶段

1. 负责人跟进整体进度
2. 小组成员定期汇报进度

### f. 模块合并阶段

1. 各成员在某个功能或页面做完之后将其合并到dev分支上
2. 所有页面都做完后对dev分支代码进行测试并合并到master分支上

### g. 测试

1. 单元测试
2. 集成测试

## 四）后台组

### a. 确定组内负责人

1. 接口文档和数据库文档的整合
2. 与其他小组定期汇报项目
3. 保持与组员的沟通
4. 跟进项目开发进度

### b. 数据库设计

1. 抽象实体，画出ER图

2. PowerDesigner建模
3. 编写数据库文档
4. PowerDesigner生成数据库

## c. 确定项目架构

1. 研究项目需求
2. 确定所需的技术与框架
3. 确定分包和编码风格

## d. 模块分工合作

1. 确定组内成员负责的模块
2. 生成JavaDoc文档提供模块接口
3. 基于版本控制系统合作及整合代码

## e. 跨域

实现前后端分离。

## f. 测试

1. Junit单元测试
2. postman接口测试
3. JMeter压力测试