1. 明确需求  
    在进行开发之前，一定先和相关人员将需求确认清楚，最好能有会议纪要，并通过邮件发送到各相关人员，后期的需求变动也需要通过邮件进行正式的确认。
2. 详细了解第三方接口  
    此项目主要需要调用华为云的接口；在需求确认的前提下，明确并记录下来需要使用到的华为云的接口；  
    对每一个要调用的接口进行请求方式、请求参数、响应参数的仔细确认；并使用工具（postman）进行一一的调用，确认每一个接口能调通；  
    找到并下载最新的SDK，找到所有要使用的接口对应的SDK中的工具类和方法；并按照SDK使用教程示例代码进行接口的调用，把每一个接口调通；  
    在使用第三方接口的过程中，有任何问题，及时发工单向第三方进行询问，电话沟通效率最高；
3. 梳理代码业务逻辑并确认项目总体结构  
    按照需求梳理出完整业务流程，如果在梳理过程中，逻辑上有模糊不清的地方，及时与需求制定人员进行明确；  
    将业务流程梳理好后，进行项目模块的划分，将业务功能划分到各个模块；进行初次建表，确认所有字段；在navicat进行初次建表，确认没有问题后，将表结构进行导出，复制粘贴到sql脚本中；
4. 代码编写  
    根据业务逻辑流程进行代码编写，每实现一个功能，就对该功能进行测试（最好使用swagger进行测试），测试通过再进行后续代码的编写；  
    代码编写中，若发现新的逻辑模糊的问题，及时与需求确认人员进行确认；尽量将所有的情况考虑进去，避免之后需求的增加，导致代码改动量大；
5. 业务流程测试  
    代码编写完成后，拉通对业务流程进行测试，自己测试通过之后，再与前端进行联调；联调通过后再交于产品进行测试。  
     
   我的问题：  
   1、前期需求不明确，写代码时有些逻辑按照自己的想法进行编写，造成后续的改动；  
     
   2、在了解了第三方的接口后，没有编写代码去测试SDK中对应的方法是否能够调通，造成后期调试时，在SDK的工具方法中发生异常；  
     
   3、在编写代码前，没有梳理详细的业务逻辑流程，造成一边编写代码一边考虑业务逻辑，编码缓慢（是否考虑使用思维导图进行业务逻辑的梳理）；  
     
   4、编写代码时，没有将业务功能划分清楚，造成一个模块完成了其他模块应该完成的业务；业务代码全部集中在controller层；没有按照一个方法只实现一个功能的思维进行代码编写，造成一个方法中代码过多，使阅读代码变得困难。  
     
   5、由于项目运行在华为云上，每次更新都需要上传镜像，重启负载，流程比较浪费时间，所以在每次更改代码后，一定要在本地测试通过再更新到华为云上。