**项目计划（简化版）**

　版本号：　1.0　　　　　制定日期：2017/6/27

|  |  |  |  |
| --- | --- | --- | --- |
| 学号 | 515030910284 | 姓名 | 于航 |
| 学号 | 515030910282 | 姓名 | 刘瑾旭 |
| 学号 | 515030910287 | 姓名 | 唐天成 |
| 学号 | 515030910292 | 姓名 | 李新宇 |
| 项目名称 | 边走边拍app | | |
| 主要的风险和应对方案：   1. 第一大风险：技术风险。   应采用什么架构，是否要采用Web service技术，本项目小组没有相关经验。   1. 缓解：请老师老师或者助教对技术方案或者软件架构进行评审，及早发现缺陷；预先开发架构原型，验证通过后才能采用。 2. 应急：咨询有经验的助教或者老师，使项目组快速学习和掌握新的知识。 3. 第二大风险：进度风险。   整个项目的开发进度较为紧迫。按照当前掌握的需求，需要一到两个月完成。但是由于可能存在的意外原因，开发时间可能还会减少一些。   1. 缓解：对需求划分优先级，采用迭代开发过程，优先级高的需求在前面的迭代实现，同时通过迭代将集成和测试提前，以保证质量。 2. 应急：当进度落后超过10%时，及时分析原因，进行改进；当进度落后超过20%时，删除一些优先级最低的需求，绝不牺牲质量。 3. 第三大风险：培训风险   所有团队成员都缺少开发经验，需要进行相关技术的学习和培训。   1. 缓解:组织项目组成员进行相关技术的学习和培训，在初始迭代中留出较为充裕的学习时间。 2. 应急：项目组及时分工，学习能力较快的小组成员进行架构开发，思维较为缜密的成员进行测试用例撰写。 3. 第四大风险：角色风险   开发人员同时担任测试角色，可能会导致更多Bug未被发现。   1. 缓解:小组成员联合进行测试，展开会议对各个模块进行测试用例撰写。 2. 应急：不同小组成员交叉进行模块的测试；设置专门的测试角色。 | | | |
| 里程碑计划：  1. 迭代1 （架构迭代）（ 6月26日 －7月6日）  缓解和应对的主要风险：技术和架构风险，培训风险  提交的成果:《项目计划》、架构原型(实现注册登录功能)、《测试用例》和《测试报告》、分析设计模型(.oom)、迭代计划（架构迭代）   * 1. 确定软件开发计划   2. 确定架构原型   3. 实现注册和登陆功能   4. 单元测试和系统测试   2. 迭代2 （第一次开发迭代）（ 7月7日 －7月15日）  缓解和应对的主要风险：进度风险，角色风险  提交的成果：第一个版本的源代码和可执行代码、演示视频文件、迭代计划（第一次开发迭代）、《测试用例》和《测试报告》  2.1 完成第一次开发迭代的编码，移交系统第一个版本  2.2 单元测试和系统测试  3. 迭代3 （第二次开发迭代）（ 7月16日 －7月23日）  缓解和应对的主要风险：进度风险，角色风险  提交的成果：测试报告，边走边拍app第二个版本  3.1 构建第二个版本  3.2 实现R2需求  3.3 测试R2需求  3.4 改进第一个版本的缺陷  4. 迭代4 （第三次开发迭代）（ 7月24日 －7月29日）  缓解和应对的主要风险：进度风险，角色风险  提交的成果：《测试用例》和《测试报告》，边走边拍app第三个版本的源代码和可执行文件  4.1 构建第三个版本  4.2 实现R3需求  4.3 测试R3需求  4.4 改进第二个版本的缺陷  5. 迭代5（产品化迭代） （ 7月30日 －8月31日）  缓解和应对的主要风险：进度风险，角色风险  提交的成果：《测试用例》和《测试报告》，可发布产品  5.1 测试产品，基于用户反馈进行微小调整  5.2 探索产品新功能 | | | |
| 项目预期成果：   * 《项目计划》 * 《迭代计划》（每个迭代开始前编写迭代计划） * 《迭代评估报告》（每个迭代结束后编写迭代评估报告） * 《SRS文档》和用例模型（.oom） * 《软件架构文档》和分析设计模型（.oom） * 《测试用例》和《测试报告》 * 《项目总结报告》 * 源代码和可执行代码 * 演示视频文件（包括安装、运行、功能等） * 演示PPT * 分工说明 | | | |