**第二次迭代计划**

　　　　　　　　　　　　　制定日期：2025/6/10

|  |  |  |  |
| --- | --- | --- | --- |
| 组号 | 72 | 项目名称 | 大学考试系统 |
| 迭代名称 | 技术原型迭代 | 计划起止日期 | 2025/5/23——2025/6/10 |
| 任务、进度安排和人员分配：   |  |  |  |  | | --- | --- | --- | --- | | No | 任务 | 起止日期 | 人员 | | 1 | 概念建模 | 6/10 | 刘黎 | | 2 | 选择语言框架并完成安装、配置 | 6/10 | 刘黎 | | 3 | 设计架构视图及关键算法 | 6/10 | 刘黎 | | 4 | 设定编程规范 | 6/10 | 刘黎 | | 5 | 撰写《软件架构文档》 | 6/10 | 刘黎 | | 6 | 搭建软件架构 | 6/10 | 刘黎 | | 7 | 实现算法或部分用例 | 6/10 | 刘黎 | | 8 | 测试技术原型并调试 | 6/10 | 刘黎 | | 9 | 编写迭代评估报告 | 6/10 | 刘黎 | | 10 | 绘制用例图 | 6/10 | 刘黎 | | | | |
| 预期成果：  1. 第二次迭代计划  2. Vision前景文档2.0  3. 软件架构文档  4. UML模型（用例模型、分析模型【用例实现，相应的vopc类图、通信图、顺序图】、设计模型【更详尽的类图】，对应的oom文件）  5. 技术原型的代码 | | | |
| 主要的风险和应对方案：   1. 进度风险：其他小组已经开始有前端成果了，而我们还停留在界面原型层次，计划时间较为紧张。如果小组成员各自写作的代码无法配合，需要调整   应对方案：制定代码规范，合理分配工作，设立监工制度，提高工作效率，争取优先实现软件框架。   1. 技术和架构风险：小组成员尚未统一学习编程语言，调试可能会遇到较大困难。   应对方案：开发过程中逐级调试，一切以运行成功为优先，尽可能减少代码修改的工作量。 | | | |