**迭代计划**

　　　　　　　　　　　　　 制定日期：2023.3.18

|  |  |  |  |
| --- | --- | --- | --- |
| 组号 | 第六组 | 项目名称 | 电影订票 |
| 迭代名称 | 技术原型迭代 | 计划起止日期 | 3.18-5.1 |
| 任务、进度安排和人员分配：   |  |  |  |  | | --- | --- | --- | --- | | No | 任务 | 起止日期 | 人员 | | 1 | 完成概念建模 | 3.18-3.20 | 安俊宇 | | 2 | 选择架构风格，  确定语言、框架、工具 | 3.21-3.22 | 全体成员 | | 3 | 设计多个架构视图 | 3.23-3.24 | 王玉宝 | | 4 | 选定编程规范，  撰写、评审软件架构文档， | 3.25-3.26 | 方嘉彤、曾永诚 | | 5 | 初步搭建软件架构 | 3.27-4.9 | 全体成员 | | 6 | 实现关键算法 | 4.10-4.16 | 全体成员 | | 7 | 继续完善软件架构 | 4.17-4.20 | 全体成员 | | 8 | 搭建测试用例 | 4.21-4.23 | 全体成员 | | 9 | 对软件进行测评 | 4.24-4.26 | 谢书洋 | | 10 | 对技术原型进行小组评审 | 4.27 | 全体成员 | | 11 | 撰写迭代评估报告 | 4.28-5.1 | 曾永诚 | | | | |
| 预期成果：  概念建模、软件架构文档、技术原型代码、2-3个测试用例、迭代评估报告。 | | | |
| 主要的风险和应对方案：   1. 技术风险   应对方案：确立实现项目的有关技术和框架；小组成员根据自身情况和发配的任务学习并掌握相关技术；采用多数同学已掌握的技术来开发软件；必要时可改变实现项目的技术，确保项目得以正常开发。   1. 进度风险： 应对方案：制定每周的迭代计划，细化近期任务；使用github进行版本管理，减少返工成本；采取基于复用、迭代的软件开发方法；如遇紧急情况应当采取补救措施（如实现某功能时遇到短期难以解决的技术问题，可果断放弃该功能）。 2. 需求风险   应对方案：进行市场调研，获取完善、准确的用户需求，并使用use-case模型描述；  通过小组讨论明确需求，对需求进行优先度排列，确立需求基线，优先满足主要需求，酌情考虑次要需求；跟踪各项需求实现情况，即时对需求进行必要的变更。 | | | |