**迭代计划**

　　　　　　　　　　　　　制定日期：2018.8.17

|  |  |  |  |
| --- | --- | --- | --- |
| 组号 | 17 | 项目名称 | HoneyComb |
| 迭代名称 | 第四次迭代 | 计划起止日期 | 8.19 – 9.9 |
| 任务、进度安排和人员分配：   |  |  |  |  | | --- | --- | --- | --- | | No | 任务 | 起止日期 | 人员 | | 1 | 合并项目后优化代码 | 8.19 – 9.8 | 朱锦昊，马轲，曹金坤，谢添翼 | | 2 | 针对使用反馈改进工程 | 8.19 – 9.8 | 朱锦昊，马轲，曹金坤，谢添翼 | | 3 | 编写测试用例，进行测试 | 9.7 – 9.8 | 谢添翼，马轲，朱锦昊，曹金坤 | | 6 | 撰写本次迭代的文档 | 9.8 – 9.9 | 曹金坤 | | | | |
| 预期成果：   1. 更加健壮的前后端分离的项目工程； 2. 完整的数据库内容； 3. 针对前后端的测试报告和用例； 4. 迭代的评估总结报告； 5. 网站的使用说明和开发接口WIKI | | | |
| 主要的风险和应对方案：   1. 组织风险：与上次迭代面临的困难类似，本次迭代开展于小学期结束后同学分散的前提下，项目组织和任务管理有较大的困难。为此，对于文档的编写我们提出更高的目标，而且为了最后的项目验收，也有必要梳理之前的文档，帮助任务收尾。 2. 需求风险：在真实测试的前提下，可能暴露之前项目开发中与实际需求不完全符合的模块设计，为此，我们需要实时追踪使用反馈，自己进行测试，保证主体需求的有效性。 3. 进度风险：诸如测试反馈再更新之类的过程，充满了不确定性，导致可能带来较大的返工需求，为此，我们应该缩短迭代中每个小更新的周期，保证较短时间内可以提供可供运行的工程版本。 | | | |