**迭代计划**

　　　　　　　　　　　　　制定日期：2024.03.27

|  |  |  |  |
| --- | --- | --- | --- |
| 组号 | 18 | 项目名称 | UniGPT |
| 迭代名称 | 技术原型迭代 | 计划起止日期 | 2024.03.28 – 2024.05.14 |
| 任务、进度安排和人员分配：   |  |  |  |  | | --- | --- | --- | --- | | No | 任务 | 起止日期 | 人员 | | 1 | 学习 prompt engineering 技巧 | 3.28 – 3.30 | 全组成员 | | 2 | 学习外部 API 调用 | 3.30 – 3.31 | 全组成员 | | 3 | 学习 spring boot | 3.28 – 4.7 | 全组成员 | | 4 | 选定或撰写编程规范 | 4.8 | 韦东良 | | 5 | 设计架构视图 | 4.8-4.12 | 黄峻涛 | | 6 | 撰写架构文档 | 4.8-4.12 | 陈启炜 | | 7 | 评审软件架构文档 | 4.13 | 全组成员 | | 8 | 学习 MySQL | 4.7 – 4.17 | 全组成员 | | 9 | 实现后端框架 | 4.12 – 4.22 | 全组成员 | | 10 | 界面原型修改 | 4.23 – 5.9 | 全组成员 | | 11 | 测试技术原型 | 5.10 – 5.12 | 全组成员 | | 12 | 编写《迭代评估报告》 | 5.10 – 5.12 | 叶懿芯 | | | | |
| 预期成果：   1. prompt engineering 文本 2. 架构视图 3. 软件架构文档 4. 外部 API 调用文档 5. 编程规范 6. 后端框架代码 7. 《迭代评估报告》 | | | |
| 主要的风险和应对方案：   1. **技术学习曲线较陡峭**：全组成员需要在短时间内掌握多项技术（如prompt engineering、外部API调用、Spring Boot、MySQL等），这可能会影响进度和质量。   **缓解措施**：使用成熟的在线课程资源。组内成员可以分工专研，之后进行知识共享。  **应急措施**：若学习进度落后，可考虑调整任务分配，优先保证核心任务的完成。   1. **技术集成和兼容性问题**：集成多种技术和外部API可能会遇到兼容性和集成困难。   **缓解措施**：在设计阶段进行技术兼容性评估和测试。选择稳定性和兼容性高的技术栈和API，使用Apifox等软件辅助测试。  **应急措施**：遇到无法解决的技术集成问题时，可寻求外部咨询助教 / 老师，或寻找替代技术方案。   1. **项目进度管理和人员协调**：团队成员可能同时参与多个任务，人员协调和进度跟踪可能会成为挑战。   **缓解措施**：使用敏捷开发方法，进行有效的进度管理和任务分配。定期召开团队会议，确保任务和进度透明化。  **应急措施**：对于关键任务落后的情况，可重新分配资源或调整截止日期，确保项目关键里程碑的达成。   1. **软件质量和稳定性**：在迭代开发的过程中，可能会因为时间压力而忽视代码质量和系统测试。   **缓解措施**：制定严格的编码标准和代码审查流程。安排足够的时间进行单元测试、集成测试和性能测试。  **应急措施**：如果发现严重的质量问题，可暂停新功能开发，优先解决现有问题。 | | | |