**迭代计划（简化版）**

　　　　　　　　　　　　　制定日期：2024.07.03

|  |  |  |  |
| --- | --- | --- | --- |
| 组号 | 1 | 项目名称 | UniGPT - v2.0 |
| 迭代名称 | 技术原型迭代 I | 计划起止日期 | 2024.07.04 – 2024.07.16 |
| 任务、进度安排和人员分配：   |  |  |  |  | | --- | --- | --- | --- | | No | 任务 | 起止日期 | 人员 | | 1 | 学习langchain4j 框架，形成新的后端重构方案 | 7.4 – 7.6 | 叶懿芯，陈启炜，黄峻涛 | | 2 | 使用langchain4j 框架重构后端 | 7.6 – 7.8 | 黄峻涛 | | 3 | 前端测试（长期） | 7.4 – 7.14 | 韦东良 | | 4 | 知识库功能后端开发 | 7.8 – 7.10 | 叶懿芯 | | 5 | 工具调用功能后端开发 | 7.8 – 7.9 | 陈启炜 | | 6 | 流式输出功能开发 | 7.9 – 7.10 | 黄峻涛 | | 7 | 知识库、工具调用、流式输出等功能的前端界面开发 | 7.11 – 7.12 | 韦东良、陈启炜 | | 8 | PostgreSQL, MongoDB 部署及开发 | 7.8 – 7.12 | 叶懿芯 | | 9 | 采用 docker 部署数据库等 | 7.11 – 7.12 | 黄峻涛 | | 10 | 学习微服务架构 | 7.12 – 7.13 | 叶懿芯 | | 11 | 初步性能测试 | 7.10 | 陈启炜 | | 12 | 插件市场后端开发 | 7.11 – 7.12 | 陈启炜 | | 13 | 测试后代码修复（长期） | 7.4 – 7.16 | 全组成员 | | 14 | 代码与文档评审，编写迭代总结报告 | 7.15 – 7.16 | 全组成员 | | 15 | 指定微服务/无服务通信接口 | 7.13 – 7.16 | 全组成员 | | 16 | 工具调用无服务探索 | 7.15 – 7.16 | 陈启炜 | | 17 | 知识库无服务探索 | 7.15 – 7.16 | 叶懿芯 | | 18 | 微服务分离架构探索 | 7.15 – 7.16 | 黄峻涛 | | | | |
| 预期成果：   1. 前后端代码 2. 迭代计划与迭代总结报告 3. Langchain4j 学习文档及实践代码 4. 前后端 API 文档 5. 微服务架构 API 文档 6. 前端测试报告 | | | |
| 主要的风险和应对方案：  **风险1：项目验收时间提前，造成进度风险**  **描述**：项目验收时间从9月中旬提前到8月10号，导致项目时间紧张。  **影响**：整体项目进度受到影响，可能无法按时完成所有任务。  **缓解措施：**  - 重新评估项目计划，优先处理关键任务。  - 增加时间投入，确保关键任务按时完成。  - 定期召开项目进度会议，及时发现并解决问题。  **应急措施：**  - 优先处理关键任务，适当舍弃非关键任务。  - 与客户沟通，争取适当的时间延长或调整验收标准。  **风险2：langchain4j 框架学习及应用的难度**  **描述：** 学习新框架并进行重构可能需要更多时间，团队成员可能遇到难以解决的技术问题。  **影响**：可能导致任务1和任务2的延迟，进而影响整个项目的进度。  **缓解措施**：  - 安排额外的时间用于框架学习，必要时可以调整其他任务的时间安排。  - 提供相关技术培训或外部专家支持。  - 建立技术问题的内部讨论和解决机制。  **应急措施**：  - 如果遇到重大技术问题无法解决，可以考虑使用备用方案，如回退到原有框架或寻找替代框架。  - 延长重构任务的时间，并重新评估后续任务的时间安排。  **风险3：人员协调和沟通不畅**  **描述：** 团队成员在多个任务中交叉工作，任务之间的协调和沟通不畅可能导致任务延迟。  **影响：** 任务1、7、13、14、15等需要多名成员协作的任务进度可能受到影响。  **缓解措施：**  - 定期召开团队会议，确保任务进度和问题得到及时反馈和解决。  - 使用项目管理工具进行任务跟踪和进度汇报，确保所有成员了解各自的工作状态和进度。  - 明确每个任务的负责人，确保任务协调和沟通畅通。  **应急措施：**  - 在出现沟通问题时，迅速召开紧急会议解决问题。  - 调整任务分配和人员安排，确保关键任务的顺利进行。  **风险4：数据库部署及开发中的技术问题**  **描述：** PostgreSQL和MongoDB的部署及开发需要较高的技术要求，可能遇到意想不到的技术难题。  **影响：** 任务8的延迟可能会影响后续功能开发和测试。  **缓解措施：**  - 提前进行技术调研，识别潜在的技术问题和解决方案。  - 安排技术专家进行指导和支持。  - 提供必要的技术培训，确保相关人员具备所需的技术能力。  **应急措施**：  - 在遇到重大技术问题时，考虑使用备选技术方案。  - 延长数据库部署及开发任务的时间，并重新评估后续任务的时间安排。  **风险5：微服务架构的探索和实施风险**  **描述：** 微服务架构的学习、探索和实施需要时间和经验，可能会遇到难以预见的架构设计问题。  **影响：**影响整体项目的稳定性和性能。  **缓解措施：**  - 提前进行微服务架构的技术调研，识别潜在的风险和解决方案。  - 安排相关人员进行培训和学习，提升技术水平。  - 建立技术支持小组，提供架构设计和实施的指导和支持。  **应急措施：**  - 在遇到重大架构问题时，考虑简化微服务设计或暂时回退到单体架构。  - 延长微服务架构探索和实施的时间，并重新评估后续任务的时间安排。 | | | |