**迭代计划（简化版）**

　　　　　　　　　　　　　制定日期：2024.07.16

|  |  |  |  |
| --- | --- | --- | --- |
| 组号 | 1 | 项目名称 | UniGPT - v2.0 |
| 迭代名称 | 第三次迭代 | 计划起止日期 | 2024.07.17 – 2024.08.08 |
| 任务、进度安排和人员分配：   |  |  |  |  | | --- | --- | --- | --- | | No | 任务 | 起止日期 | 人员 | | 1 | 分析当前版本的需求 | 7.17 | 全组成员 | | 2 | 进行微服务架构的设计 | 7.17-7.18 | 叶懿芯，陈启炜，黄峻涛 | | 3 | 微服务架构实现（Plugin User部分） | 7.19-7.29 | 叶懿芯 | | 4 | 微服务架构实现（Chat Bot部分） | 7.19-7.29 | 黄峻涛 | | 5 | Kubernetes技术学习 | 7.17-7.18 | 韦东良 | | 6 | 在云服务器上部署Kubernetes并进行应用 | 7.19-8.1 | 韦东良 | | 7 | Nginx与Redis技术学习 | 7.17-7.18 | 陈启炜 | | 8 | 在云服务器上部署Nginx与Redis | 7.19-7.29 | 陈启炜 | | 9 | 后端测试 | 7.30-8.7 | 叶懿芯 | | 10 | 插件无服务代码实现 | 7.30-8.5 | 陈启炜 | | 11 | 端到端测试 | 7.30-8.3 | 黄峻涛 | | 12 | 在云服务器上部署Kafka | 8.2-8.3 | 韦东良 | | 12 | 性能测试 | 8.6-8.7 | 陈启炜 | | 13 | 集成测试 | 8.4-8.7 | 黄峻涛 | | 14 | 在Kubernetes上部署微服务后端 | 8.4-8.7 | 韦东良 | | 15 | 代码与文档评审，编写迭代总结报告 | 8.7-8.8 | 全组成员 | | 16 | 编写项目总结报告 | 8.7-8.8 | 韦东良 | | 17 | 最终答辩PPT的制作 | 8.8 | 全组成员 | | | | |
| 预期成果：   1. 前后端代码 2. 迭代计划与迭代总结报告 3. 前后端 API 文档 4. Kubernetes部署文件 5. 项目总结报告 6. 后端测试、端到端测试、性能测试、集成测试的测试报告 | | | |
| 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：人员协调和沟通不畅**  **描述：** 本次迭代大部分位于假期时间，小组成员分散，且各自有社会实践等假期安排，任务之间的协调和沟通不畅可能导致任务延迟。  **影响：**需要多名成员协作的任务进度可能受到影响。  **缓解措施：**  - 定期召开线上腾讯会议，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3：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。  **风险4：前端和后端集成测试的问题**  **描述：** 前端和后端在功能开发完成后需要进行集成测试，可能会遇到兼容性和集成问题。  **影响：** 影响整个项目的进度。  **缓解措施：**  - 早期进行集成测试，尽早发现并解决集成问题。  - 确保前后端开发的同步进行，并保持良好的沟通和协调。  - 设立专门的集成测试小组，负责集成测试和问题解决。  **应急措施：**  - 在遇到严重的集成问题时，考虑拆分功能进行独立测试，逐步集成。  - 延长集成测试的时间，并重新评估后续任务的时间安排。 | | | |