**迭代评估报告（简化版）**

　　　　　　　　　　　　　　　　　　评估日期：2024.07.16

|  |  |  |  |
| --- | --- | --- | --- |
| 组号 | 1 | 项目名称 | UniGPT - v2.0 |
| 迭代名称 | 技术原型迭代 I | 实际起止日期 | 2024.07.04 – 2024.07.16 |
| 任务达成情况：（完成的任务、实现的功能、进度、质量等）   |  |  |  |  | | --- | --- | --- | --- | | No | 任务 | 实际起止日期 | 质量 | | 1 | 学习langchain4j 框架，形成新的后端重构方案 | 7.4 – 7.6 | 优秀 | | 2 | 使用langchain4j 框架重构后端 | 7.6 – 7.8 | 优秀 | | 3 | 前端测试 | 7.4 – 7.15 | 良好 | | 4 | 知识库功能后端开发 | 7.8 – 7.10 | 良好 | | 5 | 工具调用功能后端开发 | 7.8 – 7.9 | 优秀 | | 6 | 流式输出功能开发 | 7.9 – 7.10 | 优秀 | | 7 | 知识库、工具调用、流式输出等功能的前端界面开发 | 7.11 – 7.13 | 优秀 | | 8 | PostgreSQL, MongoDB 部署及开发 | 7.8 – 7.12 | 良好 | | 9 | 采用 docker 部署数据库等 | 7.11 – 7.12 | 优秀 | | 10 | 学习微服务架构 | 7.12 – 7.13 | 优秀 | | 11 | 初步性能测试 | 7.10 | 良好 | | 12 | 插件市场后端开发 | 7.11 – 7.12 | 优秀 | | 13 | 测试后代码修复（长期） | 7.4 – 7.16 | 优秀 | | 14 | 代码与文档评审，编写迭代总结报告 | 7.15 – 7.16 | 优秀 | | 15 | 指定微服务/无服务通信接口 | 7.13 – 7.16 | 良好 | | 16 | 工具调用无服务探索 | 7.15 – 7.16 | 优秀 | | 17 | 知识库无服务探索 | 7.15 – 7.16 | 优秀 | | 18 | 微服务分离架构探索 | 7.15 – 7.16 | 优秀 | | | | |
| 问题、变更和返工：（遇到的问题、发生的变更、是否需要返工等）   1. 问题：本次迭代新技术应用较多，课堂时间不够充足。   变更：利用课余时间进行开发，确保项目进度。   1. 问题：前端测试规模较大，但测试思路整体都类似。   变更：利用ChatGPT等大模型工具，避免过多的重复性劳动，提高效率。   1. 问题：langchain4j 框架的大多数学习资料针对的是Python版本，对Java版本仅有官方的英文文档可供学习。   变更：采用Java版本，学习框架时进行文档的翻译，遇到难题及时通过github issue向框架作者提出询问。   1. 问题：前端界面开发工作量较大。   变更：复用现有代码，例如插件编辑界面可参考机器人编辑界面的代码。 | | | |
| 经验和教训：   1. 跟进学习langchain4j这样的前沿技术并应用于项目中，使项目更具技术亮点。 2. 尝试新的技术架构之前，应当进行充分的学习和慎重的决定。 3. 参考已有的设计和功能，可以提高开发效率和减少开发的复杂度。 4. 问题和变更的发生是正常的，需要及时进行返工，确保应用达到预期的功能和效果。 5. 在开发过程中，及时进行代码评审，能够提高代码的可读性和可维护性。 | | | |