**迭代评估报告**

　　　　　　　　　　　　　　　　　　评估日期：2022年11月25日

|  |  |  |  |
| --- | --- | --- | --- |
| 组号 | 第11组 | 项目名称 | 数据流动态分析平台 |
| 迭代名称 | 迭代4构建阶段 | 实际起止日期 | 2022.11.27 - 2022.12.11 |
| 任务达成情况：（完成的任务、实现的功能、进度、质量等）  数据流动态分析平台 权限管理模块 界面原型 质量较高  数据流动态分析平台 权限管理模块 前端代码 质量一般  数据流动态分析平台 权限管理模块 后端微服务代码 质量一般  数据流动态分析平台 权限管理模块 系统测试用例 质量一般 | | | |
| 评审/测试的结果：（执行了哪些评审和测试？评审和测试的结果如何？）  **评审结果：**  经过对后端微服务架构的评审，发现开发人员的实现各有不同，迭代3中的问题仍然存在，应当继续修改架构，统一风格；  经过对数据流算子进行评审，认为应当降低算子抽象程度，降低算子抽象难度；  经过对数据流权限管理模块进行代码评审，认为代码质量符合预期，可以进行后续开发；  **测试结果：**  对后端权限管理功能进行了单元测试与集成测试，测试结果符合预期要求； | | | |
| 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）变更：对于数据流算子的抽象，在迭代3中仍然没有解决，我们需要降低算子抽象程度，采用新的方式对算子进行抽象。  2）返工：迭代3遗留的架构问题，在本次迭代中仍然存在，需要继续进行更改。代码应当采用微服务架构，使用DDD六边形架构较为规范。而早先同学们使用了分层架构，虽然可以跑通，但技术风格不统一，加大了沟通难度，造成了一定程度上的反工。 | | | |
| 经验和教训：   1. 对于项目在开发过程中遇到的技术困难，应当敏捷变更迭代计划，保证项目开发工作的正常开展； 2. 对于架构应当做好技术规范，虽然程序能够正确运行，但大大增加了沟通成本； | | | |