**迭代评估报告（简化版）**

　　　　　　　　　　　　　　　　　　评估日期：2019/1/9

|  |  |  |  |
| --- | --- | --- | --- |
| **项目组号** | 7 | **迭代名称** | 第三次迭代 |
| **实际起止日期** | 2018/12/16～2019/1/9 | | |
| **计划达成情况：**（是否按时按质完成迭代计划）  已完成：   1. 迭代计划完成的功能模块的代码编写 2. 对已完成功能模块的测试 3. 迭代所需文档的编写 4. 算法下单模块测试 5. 风控模块的开发和测试 6. 机器学习择时模块的开发和测试   未完成：  1. 无。 | | | |
| **需求实现情况：**（列出本次迭代实现的功能）  1.完成基于机器学习的交易择时策略模块  2.完成风险控制模块  3.完成算法下单模块 | | | |
| **度量结果：**  本次迭代新增或修改的源代码行数：约40000  本次迭代评审发现的缺陷数：45  本次迭代集成测试和系统测试发现的缺陷数：35  尚未修复的缺陷数：1  本次迭代投入的人月数：3\*0.7=2.1 | | | |
| **问题、变更和返工：**（遇到的问题、发生的变更、是否需要返工等）  问题：1.由于上一个迭代存在未完成的次要需求，所以占用了这个迭代的部分时间。  2.由于机器学习算法存在一定的难度，所以对需求预期做了一定的调整。  3.前后台之间交换的数据格式一度存在分歧，导致多次重新修改代码。 | | | |
| **经验和教训：**  由于上一次迭代的预期目标任务量超出预期，所以我们不得不将风险控制这一重要性比较低的需求暂行搁置，放到最后一个迭代来完成。好在我们最后一个迭代的开发计划压力比较小，时间弹性比较大，所以即使增加了任务量，也如期完成了任务。这也说明我们的迭代计划安排并不够合理，不过由于我们这个项目各模块之间耦合性比较低，所以在开发计划上的弹性比较大，但是如果项目各个模块之间依赖性比较强的话，这种不合理的进度安排很容易引起大的事故。这是一个很值得我们警醒的教训。  另外，从整体来看，我们对这个项目的预估难度还是过低，所以最后完成的产品并没有完全达到预期目标。导致这个问题的主要原因还是因为我们对量化交易这个领域完全不熟悉，所以在做项目估算的时候过于乐观，最后只能按比较低的标准来实现。这个问题给我们的体会是，相关领域专家的参与确实是项目成功的一个关键因素。 | | | |