**项目总结报告**

日期：2025年1月13日

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 组号 | 5 | 项目名称 | 集群性能监控系统及其自动化 | |
| 编程语言 | Shell | 开发平台和框架 |  | |
| **项目工作小结** | | | | |
| 1.是否实现了项目立项时的所有需求？列出实现的新增需求和未实现的需求。  2.采用哪种架构风格？哪些设计模式？  3.技术方案有哪些亮点？  4.是否做了单元测试？是否做了系统功能测试？是否做了性能测试？是否做了兼容性等其他非功能测试？  5.是否采用大模型来辅助开发？对开发生产率约有百分之几的提升？ | | | | |
| **项目组成员对项目的贡献度（%）** | | | | |
| 注：总计100% | | | | |
| **软件度量** | | | | |
| 软件代码行数（不包括注解行、空行和复用代码）： | | | |  |
| 复用他人代码行数： | | | |  |

|  |
| --- |
| **经验、教训和建议** |
| **经验:**  选择合适的技术栈: 项目团队选择了成熟的开源工具，如Prometheus、DCGM、Node Exporter和InfluxDB，这些工具经过验证，拥有丰富的文档和社区支持，降低了开发风险。  面向对象开发方法: 采用UML进行系统建模，有助于团队成员理解项目架构和流程，提高开发效率和代码质量。  迭代开发: 将项目分解为多个迭代，每个迭代解决特定的风险，逐步完善系统功能，降低了项目失败的风险。  测试驱动的开发: 使用PyTest等测试工具，确保代码质量和功能稳定性。  **教训:**  需求风险: 项目初期需要充分与用户沟通，明确需求，避免后期需求变更带来的开发成本增加。  技术风险: 技术选型需要谨慎，避免选择过于复杂或难以驾驭的技术，导致项目延期或失败。  人力风险: 需要评估团队成员的技术水平和经验，避免因技术不熟练导致开发进度延误。  时间风险: 项目管理需要严格，制定合理的开发计划，并定期进行进度跟踪和调整。  安全风险: 需要重视系统安全性，采取必要的安全措施，防止数据泄露。  **建议:**  加强需求管理: 建立需求变更流程，确保需求变更得到有效控制。  进行技术预研: 在项目启动前，对选定的技术进行充分的调研和评估，确保其适用性和可行性。  提升团队能力: 为团队成员提供必要的技术培训，提高其技术水平和工作效率。  加强项目管理: 使用项目管理工具，如Jira或Trello，跟踪项目进度，并定期进行项目评估和调整。  重视安全设计: 在系统设计阶段就考虑安全性，并进行安全测试，确保系统安全可靠。  关注用户体验: 进行用户测试，收集用户反馈，并根据反馈进行系统改进。 |

项目组各成员签字：