

项目管理方法

@吴城

适用人群

- 直接项目领导者

管理方法

- 明确最终目标和时间点
- 拆分里程碑和时间点
- 掌控项目宏观方案和模块细节
- 拆分和分配任务
- 充分频繁地沟通问题
- 定期验收
- 保留关键文档
- 培养接替人

明确最终目标和时间点

- 最终目标是什么？
 - 清晰明确（几句话就能描述清楚）（举例：精准告警，自动定位，故障自愈）
 - 组内共识（禁止沟通，写在纸上）
- 成果展望，详尽真实
- 任何项目必须要有GuideLine

拆分里程碑和时间点

- 里程碑数量不宜过多，5个以内，最好3个以内
- 每个里程碑的目标是什么？清晰明确，组内共识。举例
 - Q2：完成Open-Falcon整合
 - Q3：生成环境使用，并优化异常检测方案(同环比(流量)，曲线平滑(iops))
 - Q4：部分场景故障自愈（故障磁盘自动格式化上线，发布代码自动回滚，服务异常自动剔除慢速机器或集群）
- 成果展望，详尽真实
- 每个里程碑都必须有GuideLine

掌控项目宏观方案和模块细节

- 宏观方案是什么？架构是怎么样的？是否合理？有无风险？扩展性如何？
- 模块细节
 - 模块功能是什么？方案是否合理？有无风险？扩展性如何？
 - 若组员能力很强，只需要掌控关键模块细节
 - 若组员能力不够，则需要掌控绝大部分模块细节

拆分和分配任务

- 拆分任务的原则：独立，少耦合
- 分配任务
 - 按照能力分配
 - 明确原则，适当授权（原则：这件事我完全可以放手让他去做，只要他。。。）
- 里程碑，任务和开发过程中的bug全部用工具记录（Gitlab）
 - Milestone（4.1.1-服务树和UIC开发-0209~0306）
 - Issue（通过tag标识bug, feature）
 - 分配给指定人
 - comment中记录关键问题讨论结果或者会议纪要

充分频繁地沟通

- 大会若干，小会不断（周三，周五）
- [可选]每日例会，快速沟通进度和问题（早/晚）
- 每日关闭或者调整Gitlab中的任务
- 不断传播和共识最终的或者里程碑的目标和成果
展望

定期验收

- 至少一周两次验收，周三，周五

保留关键文档

- 方案评估文档
- 系统设计文档（逻辑，原理，架构，核心算法）
- 接口文档
- Q&A（常见问题，关键部分的思考过程）
- Troubleshoot文档

培养接替人

- 越多人知道全局，越能够把项目做好
- 项目规模大时，分担一部分管理工作
- 自己请假时能够帮忙接替工作（往下委托或者评级委托）

Q & A